Analyzing Candidate Speaking Time in Estonian Parliament Election Debates

Siim Talts and Tanel Alumäe[0000−0001−5083−1556]

Tallinn University of Technology, Estonia
siim951@gmail.com, tanel.alumae@taltech.ee

Abstract. In this paper, we analyze the amount of speaking time by each candidate and political party during the election debates that aired in broadcast media during the Estonian 2019 parliament election campaign, using automatic speaker identification. We use automated methods for retrieving speech recordings from publicly available sources that are likely to contain speech by the target speakers, and apply a weakly supervised method for training speaker recognition models from such data. The experiments show that the resulting models have high precision but they lack in recall. While most candidates from large and established parties could be identified automatically, candidates from smaller and newer parties could not be recognized, due to their little prior media presence. Our system allows to identify candidates who spoke for the longest time in the election debates. The analysis shows that the election debates were not biased from the speaking time point of view: all major political parties received relatively similar speaking time across the debates.

Keywords: Elections, Political Debates, Speaker Recognition, Estonia.

1 Introduction

Publicly aired debates between election candidates are an important part of a well-functioning democratic process as they increase political accountability and voter knowledge. Debates have the potential to provide valuable information to citizens on the character and policy positions of politicians. Empirical evidence suggest that voters acquire significant political knowledge from watching and listening to the debates [4]. This knowledge has been found to persist over a number of weeks, and influence the voting choices on the election day.

Previously, computational models have been used for detecting claims in political debates, [9, 12, 16], analyzing non-verbal expressions [10] and performing automatic fact-checking [8]. The problem of automatically identifying speakers in election debates has not been investigated before. Yet, recognizing who is speaking at a particular moment in the debate can be useful for several purposes. For example, this makes it possible to produce an archive of political debates that is automatically annotated and indexed at a speaker lever. Such archive would allow to easily retrieve segments from a large library of political debates where a given politician is speaking, in order to detect political claims or deception. Automatic speaker recognition could be also used for

studying the balancedness and possible bias of mass media during the election campaign.

This paper has two goals. First, we investigate whether it is possible to create speaker recognition models for identifying election candidates in political debates using no hand-annotated training data. We use automated methods for retrieving speech recordings from publicly available sources that are likely to contain speech by the target speakers, and apply a weakly supervised method for training speaker recognition models from such data. Second, we apply the resulting models on Estonian parliament election debates and analyze the speech duration over individual candidates and political parties.

This paper is organized as follows: in the next section, the weakly supervised method is described. Chapter 3 summarizes the Estonian parliament election system and describes the election campaign and coverage. Chapter 4 gives details about the data collection procedure, model training, model validation and describes the findings of our experiments. Paper ends with conclusion and suggests some ideas for future work.

2 Training Speaker Identification Models

2.1 Background

Conventional fully supervised text-independent speaker identification training methods require manually segmented training data: for each person that the system needs to identify (often called person-of-interest, or *POI*), we need several speech segments where only this person is speaking (Figure 1, left). The segments should cover different speaking styles (e.g., interview, public speaking), acoustic channels (microphone, telephone) and acoustic conditions (studio, background noise). Based on such hand-segmented training data, a speaker identification model learns a fixed-dimensional embedding of the person's voice. The embeddings are usually computed through factor analysis, resulting in i-vector based embeddings. More recently, deep neural network based speaker embeddings, or x-vectors, have been getting more popular due to their higher accuracy. At test time, embedding calculated from a speech segment corresponding to an unknown speaker is compared to the embeddings of all POIs. Simple cosine similarity or a more complicated discriminative classifier, such as a probabilistic linear discriminative analysis (PLDA) model, can be used for comparing the embeddings. If the difference between a test segment and a POI is lower that a tuned threshold, the speech segment is assigned to the given POI.

Producing manually segmented training data for fully supervised speaker identification is costly. If we wanted to train a model with wide coverage (e.g., thousands of politicians for media monitoring purposes), we would need to find several speech recordings where the given POI appears, and manually mark the segment boundaries where this POI is speaking. This requires a lot of human labour, and it is difficult to require such system up-to-date, as it requires annotating additional data (e.g., to cover new POIs).

Recently proposed weakly supervised [7, 2] training relaxes the training data needs by only requiring recording-level labels (Figure 1, right). That is, for each POI, we

Fig. 1. Supervised (left) and weakly supervised (right) training data for speaker identification. In the supervised scenario, speech segments of the enrolled persons are explicitly annotated. In the weakly supervised case, only recording-level speaker labels are required while time-based annotation is not needed.

still need several recordings where this POI appears, but we don't need to annotate the data at the segment level. Instead, for each recording in the training set, we need to provide a set of POIs who appear there. This makes training data collection and annotation much easier. Furthermore, sometimes the information about the speakers appearing in the recording is given in the metadata. In such case, we would just need to find recordings to cover all POIs based on the recording metadata, and produce the corresponding recording-level labels based on the metadata.

2.2 Weakly Supervised Training

The method that we first proposed in [7] requires a dataset of audio recordings and the corresponding recording-level speaker labels. The speaker labels represent the sets of POIs whose speech appears at least once in the recording. The set of speakers does not need to be exhaustive: only the speakers that need to be identifiable by the system (POIs) must be included in the sets.

Outline of the training process is depicted in Figure 2.

First, we apply a speaker diarization [15, 5] and i-vector [6] extraction system to the training data (Figure 3). This step partitions a recording into homogeneous segments, discards non-speech segments and clusters the speech segments that are likely uttered by the same speaker. Next, we use an i-vector extractor to compute i-vectors for all speakers in all recordings. I-vectors allow to map variable-length speech utterances to fixed-dimensional vectors. In speaker recognition, the dimensionality of the i-vectors is typically in the range of 400 to 600. The goal of i-vectors is to map similar utterances (i.e., those from the same speaker) to similar i-vectors. I-vectors are based on generative probabilistic model using Gaussian Mixture Models. I-vectors are widely used for speaker, language, dialect and gender recognition, and they have also been used for identifying various paralinguistic phenomena, such as emotions. The i-vector extractor can be trained on some available speaker-labeled training data, possibly from another domain. Alternatively, the i-vector extractor can be trained on the automatically clustered speakers of the training set. We experimented with both methods and found no clear difference between those alternatives.

Here we give some useful notations for the rest of this section. Let $\mathscr{X} \subset R^D$ denote the D-dimensional feature space (i.e., the i-vector space) and $\mathscr{Y} = \{1, 2, ..., C\}$ the set

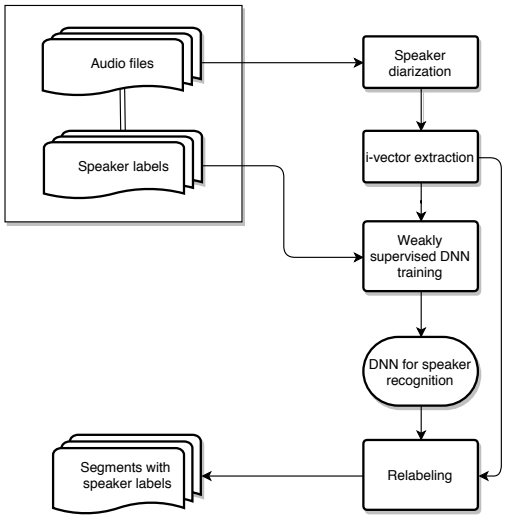

Fig. 2. Architecture of the weakly-supervised training process.

of target speaker identities. The training corpus $\mathscr{D}$ contains a set of N audio recordings $\{X_n\}_{n=1,2,\ldots,N}$ where each recording X_n contains a set of i-vectors for the diarized speakers $X_n = \{x_{mn}\}_{m=1,2,\ldots,M_n}$, with $x_{mn} \in \mathscr{X}$. The training corpus also contains the corresponding sets of speaker labels for each recording $\{Y_n\}_{n=1,2,\ldots,N}$ where $Y_n \subset \mathscr{Y}$. Note that the number of speaker labels does not have to agree with number of i-vectors for the corresponding recording, and the correspondence between x_{mn} and the elements in $\{Y_n\}$ is not known. The task is to train a model to classify i-vectors based on the speaker identities, i.e., to learn a classification function $f : \mathscr{X} \rightarrow \mathscr{Y}$ from the training dataset.

We use a feed-forward neural network to learn the mapping from the i-vector space to the speaker label space. First, we augment the set of target speaker identities with a special $\langle unk \rangle$ identity reserved for unknown speakers, $\mathscr{Y}' = \mathscr{Y} \cup \{\langle unk \rangle\}$. The neural network takes i-vectors as input and has $|\mathscr{Y}'|$ outputs. The softmax function is used in the final layer of a neural network.

The neural network is trained using an objective function defined at the recording level, not at the single training sample level as usually. Specifically, we want the neural network to predict a similar set of speakers for the set of i-vectors as is given in the label set. To achieve this, we first define the expected average distribution over the speaker labels for each recording as

$$\bar{\tilde{p}}_n(y_i) = \begin{cases} \frac{1}{|X_n|}, & if\ y_i \in Y_n \\ max(0, 1 - \frac{|Y_n|}{|X_n|}), & if\ y_i = \langle unk \rangle \\ 0, & otherwise \end{cases} \quad (1)$$

The recording level objective function is the Kullback-Leibler divergence between the

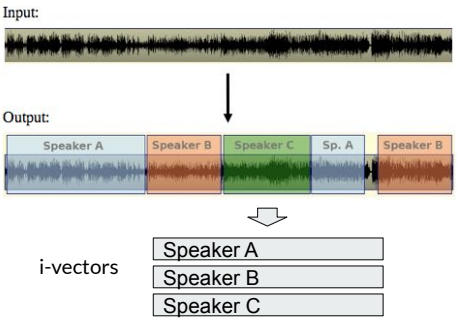

Fig. 3. Speaker diarization and i-vector extraction: speech recording is segmented into homogeneous segments; segments that contain similar-sounding speech are clustered. For each cluster (i.e., likely a unique speaker), a fixed-dimensional representation (i-vector) is extracted.

expected average distribution and model's expected average conditional distribution:

$$D(\bar{\tilde{p}}||\bar{\hat{p}}_\theta) = \sum_y \bar{\tilde{p}} \log \frac{\bar{\tilde{p}}}{\bar{\hat{p}}_\theta} \tag{2}$$

The idea of this objective function is that we don't know the exact correspondence between the i-vectors and speakers of a recording, but we want exactly a single i-vector to be assigned to each speaker of the show, and the rest (if any) to be absorbed by the class that is reserved for unknown speakers. Clearly, the assumption that only one i-vector corresponds to a labeled speaker is not always true: in broadcast news, a news anchor could be speaking at the beginning of the news show with background music, and later without music, which usually causes the speaker diarization module to split the single speaker into two pseudo-speakers. Similarly, a reporter could speak both in a studio setting and with a noisy background in a single news show, also resulting in two i-vectors. However, we haven't found this to cause any noticeable problems.

The proposed method works only if the recordings in the training data have sufficiently different speaker distributions. The objective function does not explicitly encourage the neural network to assign a high probability to a particular speaker in the recording – given a single recording and a set of speakers for that recording, the objective function can be minimized by predicting a uniform distribution over all i-vectors for all speakers appearing in that particular recording. However, since we require the recordings to have different speaker distributions, the neural network has to start assigning non-uniform distributions to the i-vectors of a show, in order to better fit all the data.

The neural network also doesn't learn to differentiate between the speakers that occur always together in the same recordings.

The training algorithm is summarized in listing 1.

We used a very simple deep neural network as the underlying model that maps i-vectors to speakers (Figure 4). The network has two fully-connected hidden layers, using the leaky rectified linear units. The number of hidden units was optimized on

Data:
List of recordings, diarized, i-vectors extracted;
Set of POIs for each recording;
Result: Trained model that maps i-vectors to POIs
begin
  Precompute expected average speaker distribution $\bar{\tilde{p}}_n$ for each recording according to eq. 1;
  Randomly initialize the model;
  while *training hasn't converged* **do**
    Shuffle training data;
    for *each recording* **do**
      Compute DNN posteriors for all i-vectors X_{nj} of recording n according to the current model;
      Average the predictions over the recording;
      Compute KL-divergence between the predicted and expected average distributions;
      Compute gradients and update the model;
    end
  end
end

Algorithm 1: Algorithm for training a neural network on weakly labeled data.

development data. We found that dropout layers added after the dense layers improve performance of the model.

Once the speaker recognition DNN is trained, it can be used directly for speaker identification, as we proposed in [7]. It can be also used for relabeling all segments found during speaker diarization, resulting in new segment-level annotations for the whole dataset. Since the DNN has intrinsically a dedicated output for the unknown speaker class, we can simply discard segments that are classified to the unknown speaker. The resulting annotations can be used for performing LDA/PLDA based speaker recognition.

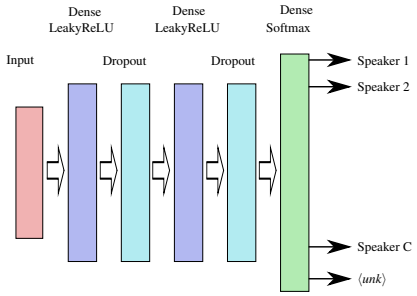

Fig. 4. Architecture of the speaker identification DNN.

3 Estonian Parliament Elections

3.1 Estonian Parliament

Parliamentary elections were held in Estonia on 3 March 2019. The parliament *Riigikogu* has 101 seats. Members are elected by proportional representation in 12 constituencies. Seats are allocated using a modified D'Hondt method. Parties have to get at least 5% votes nationwide to be represented in the parliament. Individual candidates can get elected by exceeding a simple quota in their constituency (obtained by dividing the number of valid votes cast in the electoral district by the number of seats in the district). The remaining seats are allocated based on each party's share of the vote and the number of votes received by individual candidates. The remaining seats are filled using a closed list presented by each party at the national level [1].

3.2 Candidates

There were 1084 candidates in the 2019 elections from ten political parties. In addition, there were 18 independent candidates. Each party could nominate a maximum of 125 candidates (so called full list), divided into 12 constituencies. Eight parties participated in the elections with the full list while two parties had less than 125 candidates.

3.3 Election Campaign

Election campaign starts typically several months before the election date. According to the Estonian law, the active election campaign period starts 45 days before the elections. During that time, political advertisements are forbidden in open public space (e.g., billboards). Other forms of campaign, such as paid TV and radio clips, adverts in social media, campaign events and person-to-person campaigns, are allowed.

An important part of the election campaign are debates on TV and radio. Election debates are aired by both public as well as private broadcasters. Election debates are usually focused on a specific topic (e.g., social welfare). Usually, the broadcaster decides which parties to invite to the debate, and the party decides on the specific candidate that represents them.

4 Analysis of Candidate Speaking Time

4.1 Collection of Training Data for Weakly Supervised Training

In order to train speaker identification models using the weakly supervised method described in section 2.2, we need to collect training data for the candidates. Training data should consists of audio recordings, labeled with the names of the candidates whose speech appears at least once in the recording.

Since the number of candidates is high, we decided to use an automated method for collecting the data. We scraped audio data from two sources: archive of the Estonian Public Broadcasting (*ERR*) and YouTube.

Table 1. Training data scraping result

	# recordings	# speakers
ERR archive	6619	545
YouTube	4526	733
Total	**11145**	**810**

Collecting data from the *ERR* archive is relatively straightforward. We relied on the metadata of the radio broadcasts in the archive that for most recordings lists the names of the persons appearing in the recording. We developed an aggregator that firstly creates an indexed map of all the recordings in the archive by using metadata available. This is done by looping through a range of pages in the archive and collecting all the data that is presented in the metadata section of that page. Then, the recordings which contain one of the candidates are downloaded. Several candidates can occur in the same recording.

For YouTube, a different approach had to be taken. We used the YouTube API to search for videos that contain the candidate's name either in the title or the video metadata. It is obvious that this kind of metadata is less reliable, as the fact that a person's name appears in the title or metadata does not guarantee that the person is speaking in the video.

The results of the data scraping are listed in Table 1. Scraping was ran for 1084 candidates which resulted in over 550 GB of training data with 11145 unique recordings. The data was collected in January 2019. The goal was to collect training data for most candidates. As can be seen in Table 1, we were able to scrape training data for 810 names. However, the weakly supervised training method requires each speaker to occur in multiple recordings. When setting a minimum of 10 recordings per candidate, only 317 of the candidates remained, meaning that we were able to create models for only about 30% of them.

As can be expected, the number of scraped recordings per candidate varies a lot. The two candidates with most training data were Taavi Rõivas (374 recordings) and Jüri Ratas (342) – former and current prime minister, respectively. Most of the other persons in the top are experienced politicians who speak often on the radio. Figure 5 shows the number of recordings for all candidates with at least 10 items (note that only every 10th name is given).

In order to assess the precision of data scraping procedure, we randomly selected 50 recordings from the scraped pool and manually verified whether a candidate that the recording corresponds to actually appears there. Based on the sample, data from *ERR* has 100% precision while YouTube data precision is 73%.

4.2 Training Speaker Identification Models

In order to train speaker identification model on the scraped data, we used the speaker diarization system developed for the Estonian rich transcription system [3]. The diarization module is based on the LIUM SpkDiarization toolkit [11]. Kaldi [14] was used for extracting i-vectors for the speaker clusters in the diarized data. The weakly supervised training algorithm was implemented in Pytorch [13].

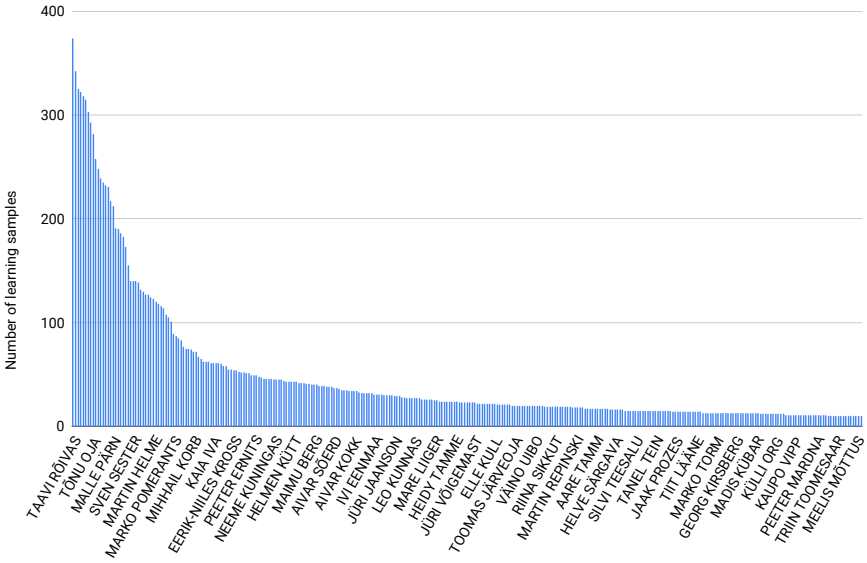

Fig. 5. Training recording count by candidate.

4.3 Model Validation

In order to validate that the trained model is accurate, we randomly chose four election debates and manually segmented and labeled them at the speaker turn level. In total, the validation dataset consisted of 5 hours and 32 minutes of audio and it included 26 different candidates.

From among the 26 different candidates in the validation data, a speaker identification model was trained for 21 (78%) of them. For the rest of the five candidates, not enough data was scraped in order to train speaker identification models.

The trained system was able to identify 19 candidates of the 21 covered by the models. Both of the false negatives were probably created due to the shortage of training samples – one of the speakers had 13 and the other 10 samples. There were no false positives which means that the system did not identify anybody who wasn't actually speaking in the debates.

4.4 Test Data

After the trained model was validated, the main experiment for estimating candidate speaking time in election debates could be executed. For this, we manually picked and downloaded election debates from six different TV and radio stations. In total, the test dataset consisted of 55 different recordings with nearly 55 hours of data. Based on the metadata of the the debates, the recordings had, not including the presenters, 210

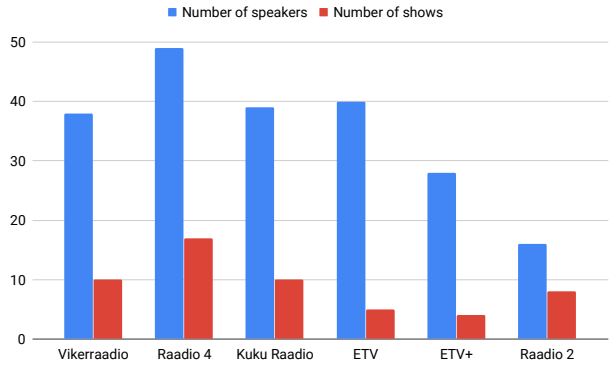

Fig. 6. Number of debates and unique speakers per station.

speakers. The number of unique speakers over all the test data was 123. The number of downloaded recordings and the number of unique speakers in those debates per TV/radio station is given in Figure 6.

4.5 Speaking Time Results

The test data was processed by a speaker diarization system that splits the recordings into utterance-like segments and clusters the segments based on the speaker. Then, the trained speaker recognition model was applied to the i-vectors extracted from each resulting cluster.

Firstly, we evaluated the overall recall and precision of the speaker recognition system. Based on the recording metadata, there were 123 unique speakers in the test data. Speaker recognition model was available for 69 (56%) of them.

Next, we summed the total speaking time of each candidate that the speaker recognition model could identify. In the post-election analysis, we also collected the number of votes each candidate retrieved. Figure 7 shows the candidates with the most amount of detected speaking time in the election debates, together with the number of votes received.

It is not surprising the top seven places in terms of total speaking time are all occupied by party leaders. However, there are are large differences in the votes that they received which are well correlated with the election result of the corresponding political party. The number of votes received by different candidates are however not directly comparable to each other since the constituencies are of different size.

We do not claim that our results show any causal effect between the amount of speaking time of the candidate and the number of votes received. Although there is probably a positive correlation between the two numbers, there are several factors, such as prior popularity, speaking skills and experience in political debates, that affect both exposure in debates as well as the number of votes received.

We also analyzed the results in terms of total speaking time of the political parties. When we simply sum up the detected speaking time of the party's candidates, then large

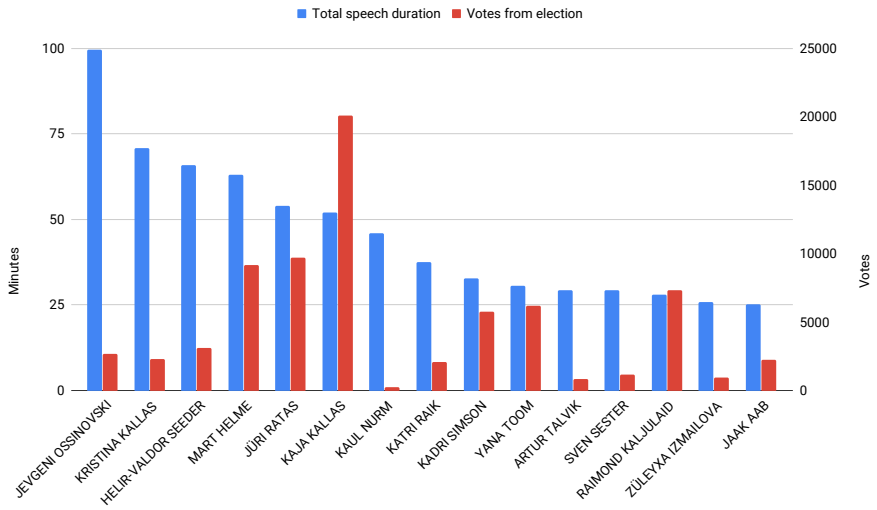

Fig. 7. Candidates with the most amount of speaking time, together with the votes received.

differences between the parties can be observed (Figure 8, purple bars): *SDE* candidates spoke for over 250 minutes, while *Rohelised* only around 50 minutes. However, such large difference is mostly due to the weakness of our model training process: candidates of large and established parties have had more exposure on TV and radio and in YouTube, and were therefore more likely to have enough automatically collected training material. Smaller parties (*Eesti 200*, *Vabaerakond*, *Rohelised*, *Elurikkuse Erakond*, *Vasakpartei*) had a lot of candidates who are new to politics and have almost never appeared on TV/radio before.

In order to estimate speaking time by parties more robustly, we computed two additional speaking time estimates. First, we tried to estimate a speculative speaking time duration of those candidates who were not detected for a particular debate. Since we knew the set of candidates appearing in each debate, we computed the mean speaking time for the detected speakers for each recording, and assigned the same mean to the undetected speakers. When taking this account, the amount of speaking time for different parties is much more uniform (Figure 8, gray bars). Only the smallest parties still have considerably less exposure. This can be explained by election coverage rules imposed by *ERR* which mandated that only the parties that participate with a full list of candidates (125 persons) are invited to all debates.

We also computed a second speculative per-party speaking duration where we only relied on the metadata and did not use the speaker detection results at all: here we simply divided the total debate running time (multiplied by 0.8 to take the speaking time of debate moderators into account) between the persons appearing in that debate, based on the metadata (Figure 8, blue bars). We can observe that there is relatively little difference between the two estimated speaking times. This suggests that our speaker

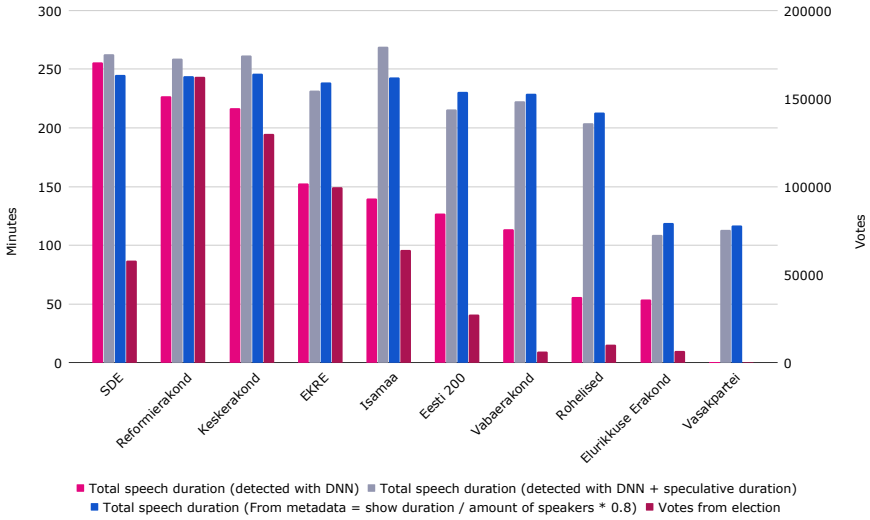

Fig. 8. Total speaking time and votes received for political parties.

recognition model does not really give any benefit for comparing speaking time between the parties, since it is easier to simply rely on the debate metadata.

5 Conclusion

Our results indicate that it is possible to train an ad-hoc speaker recognition system with high precision simply by scraping the internet for audio recordings, given that the target speakers have enough exposure on public media and video-sharing platforms, such as YouTube. We used this method to build a speaker identification system for the candidates of the Estonian parliament elections. Results show that while the system can accurately identify experienced politicians, the recall for candidates from newer and smaller parties is relatively low due to their little previous media presence.

When accounting for the weaknesses of our models, the analysis shows that media coverage of the Estonian 2019 parliament election debates was not biased from the speaking time point of view: the total speaking time of different political parties was similar.

There are some avenues for future research. First, our study concentrated only on the election debates. In addition, candidates also appear in TV/radio ads and broadcast news. This could possibly have a larger impact on the success of the candidate than performing in election debates, as a significant part of the population does not watch election debates. Second, if we want to further analyze the possible media bias, then the amount of speaking time is probably not the most significant factor, as the general sentiment and communication of the debate moderators could play a larger role. Automatic speech recognition and emotion recognition can be applied for this kind of analysis.

References

1. Riigikogu election act. Riigi Teataja (January 2015).
2. Alumäe, T.: Training speaker recognition models with recording-level labels. In: Proc. SLT. pp. 1066–1072 (2018).
3. Alumäe, T., Tilk, O., Asadullah: Advanced rich transcription system for Estonian speech. In: Proc. Baltic HLT (2018).
4. Bidwell, K., Casey, K., Glennerster, R.: The impact of voter knowledge initiatives in Sierra Leone. Policy brief (2014).
5. Chen, S., Gopalakrishnan, P.: Speaker, environment and channel change detection and clustering via the Bayesian information criterion. In: Proc. DARPA Broadcast News Transcription and Understanding Workshop (1998).
6. Dehak, N., Kenny, P.J., Dehak, R., Dumouchel, P., Ouellet, P.: Front-end factor analysis for speaker verification. IEEE Transactions on Audio, Speech, and Language Processing **19**(4), 788–798 (2011).
7. Karu, M., Alumäe, T.: Weakly supervised training of speaker identification models. In: Proc. Odyssey 2018 The Speaker and Language Recognition Workshop. pp. 24–30 (2018).
8. Kopev, D., Ali, A., Koychev, I., Nakov, P.: Detecting deception in political debates using acoustic and textual features. In: Proc. ASRU (2019).
9. Lippi, M., Torroni, P.: Argument mining from speech: Detecting claims in political debates. In: Proc AAAI (2016).
10. Maurer, M.: Nonverbal influence during televised debates: Integrating CRM in experimental channel studies. American Behavioral Scientist **60**(14), 1799–1815 (2016).
11. Meignier, S., Merlin, T.: LIUM SpkDiarization: an open source toolkit for diarization. In: Proc. CMU SPUD Workshop (2010).
12. Menini, S., Cabrio, E., Tonelli, S., Villata, S.: Never retreat, never retract: Argumentation analysis for political speeches. In: Proc. AAAI (2018).
13. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems 32, pp. 8024–8035 (2019).
14. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., Vesely, K.: The Kaldi speech recognition toolkit. In: Proc. ASRU (2011).
15. Reynolds, D.A., Singer, E., Carlson, B.A., O'Leary, G.C., McLaughlin, J.J., Zissman, M.A.: Blind clustering of speech utterances based on speaker and language characteristics. In: Proc. SLP (1998).
16. Rohit, S.V.K., Singh, N.: Analysis of speeches in Indian parliamentary debates. arXiv preprint arXiv:1808.06834 (2018).