See me Speaking? Differentiating on Whether Words are Spoken On Screen or Off to Optimize Machine Dubbing

Shravan Nayak
pshravan.nayak.ece17@itbhu.ac.in
Indian Institute of Technology (BHU)
Varanasi

Timo Baumann
baumann@informatik.uni-hamburg.de
Universität Hamburg

Supratik Bhattacharya
f20170745@pulani-bits-pilani.ac.in
Birla Institute of Technology and Science, Pilani

Alina Karakanta
akarakanta@fbk.eu
Fondazione Bruno Kessler / University of Trento

Matteo Negri
negri@fbk.eu
Fondazione Bruno Kessler

Marco Turchi
turchi@fbk.eu
Fondazione Bruno Kessler

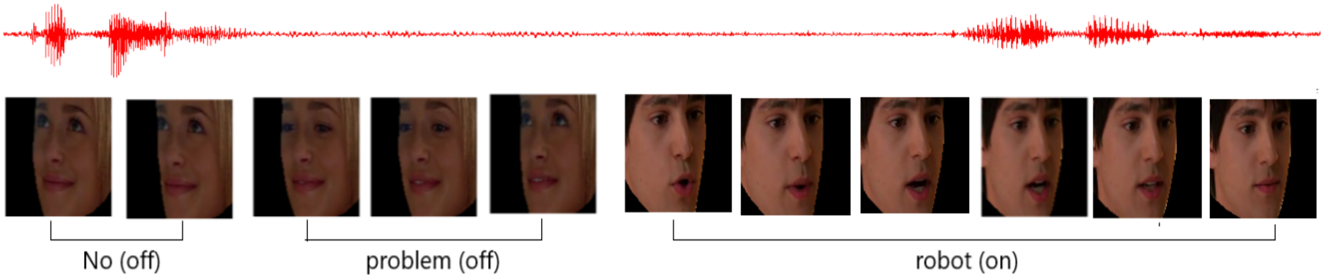

Figure 1: An example: "No problem, robot", of the classification task along with the annotations (s2_1_250).

ABSTRACT

Dubbing is the art of finding a translation from a source into a target language that can be lip-synchronously revoiced, i. e., that makes the target language speech appear as if it was spoken by the very actors all along. Lip synchrony is essential for the full-fledged reception of foreign audiovisual media, such as movies and series, as violated constraints of synchrony between video (lips) and audio (speech) lead to cognitive dissonance and reduce the perceptual quality. Of course, synchrony constraints only apply to the translation when the speaker's lips are visible on screen. Therefore, deciding whether to apply synchrony constraints requires an automatic method for detecting whether an actor's lips are visible on screen for a given stretch of speech or not. In this paper, we attempt, for the first time, to classify on- from off-screen speech based on a corpus of real-world television material that has been annotated word-by-word for the visibility of talking lips on screen. We present classification experiments in which we classify individual words as on- or off-screen. We find that this task is far from trivial and that combined audio and visual features work best.

CCS CONCEPTS

• **Computing methodologies → Activity recognition and understanding**; **Machine translation**.

KEYWORDS

audiovisual machine translation ; dubbing ; multi-modal speech processing ; activity recognition

ACM Reference Format:
Shravan Nayak, Timo Baumann, Supratik Bhattacharya, Alina Karakanta, Matteo Negri, and Marco Turchi. 2020. See me Speaking? Differentiating on Whether Words are Spoken On Screen or Off to Optimize Machine Dubbing. In *Companion Publication of the 2020 International Conference on Multimodal Interaction (ICMI '20 Companion), October 25–29, 2020, Virtual event, Netherlands.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3395035.3425640

1 INTRODUCTION

Dubbing is a form of audiovisual translation (AVT) [11] which consists in replacing the original speech of a film with another track in a different language to make it accessible to international audiences. The goal is to make it appear as if the film had been recorded in the target language all along. Therefore, translation for dubbing needs to follow the timing and phrasing of the original language, and special attention needs to be paid to matching phonetic features

ICMI '20 Companion, October 25–29, 2020, Virtual event, Netherlands

ACM ISBN 978-1-4503-8002-7/20/10...$15.00
https://doi.org/10.1145/3395035.3425640

such as lip closure and opening angle of the jaw [4]. This additional attention on visual-phonetic matching poses an extra constraint for the translation and often results in wordings that are not a literal translation of the source language. Dubbing is the preferred type of video translation in countries with large film and streaming markets, e. g. Germany, Italy, Spain and parts of Latin America.

Previous work on automatic machine dubbing has focused mostly on matching the length between source and target utterances, either in terms of characters [5] or syllables [14], and has shown that, while translation quality degrades only moderately, introducing such matching constraints improves 'dubbability' (i. e., how well the translations can be spoken synchronously to the original speech). This previous work, however, ignores for simplicity whether speech is visible on screen or not, although 'dubbability' only applies to speech that is spoken by a face visible on screen. In this paper, we tackle the problem of automatically identifying those stretches of speech where the actors' lips are visible on screen. As translation quality is negatively affected when applying additional constraints, automatically identifying the stretches which require lip synchrony is an important contribution towards avoiding drops in translation quality for segments where no dubbing constraints are required.

Perhaps most similar to our work, Roth et al. [13] present a large dataset (AVA-ActiveSpeaker) and models for detecting the speaker in a video. The main difference is that they annotate every face in every frame of the video clip whereas we only have word-level annotations which we later align to the video. This means that when multiple faces are visible on the screen for an on-screen word we have no ground truth as to which face is speaking but only the fact that a word is being spoken on screen. Furthermore, the time duration of our video clips ranges from 1–10 s (whereas video duration is fixed in [13]). This shows that the task of automatically detecting which parts of a video require lip synchrony is not a trivial task that can be solved with currently available off-the-shelf speaker recognition solutions.

2 METHODOLOGY

2.1 Dataset

We use the television series Heroes which has previously been transcribed and made available as a corpus [10] and which is, to our knowledge, the only freely available dubbing corpus of real-world television.[1] The corpus consists of 7000 English utterances and their corresponding Spanish dubbed utterances. As the original corpus is published without the video, we have added video from the DVD release (25 frames/s) to provide for multi-modal analysis.

2.2 Annotation

Our aim is to classify whether a word is spoken on- or off-screen, in order to allow for fine-grained application (or relaxation) of dubbing constraints; in particular we note that a large proportion of all utterances in the corpus are a *mixture* of on- and off-screen words, e. g. when a cut changes the camera from speaker to listener. For this, we manually annotated all spoken words in the corpus, as to whether they are spoken on- or off-screen (40314 resp. 16054). A first annotator went through all data, and a second annotated 700 randomly selected utterances to check the reliability of annotation. This resulted in utterances which were either fully on-screen (4163), fully off-screen (1200) or a mixture of on-screen and off-screen words (1614). The two annotators had an inter annotator agreement (Cohen's Kappa) of $\kappa = .73$ for the word-level annotation. Hence we conclude that on/off annotation can be done with substantial agreement among the annotators. We find that most differences appear for utterances that contain a mixture of on-screen and off-screen words, where the boundary between on and off differs by a few words between the annotators.

2.3 Data Preparation

Preparing the dataset consisted of three main stages: 1) aligning the text, speech and video, 2) face detection and extraction, and 3) audio feature extraction, as detailed below.

2.3.1 Text-Speech-Video Alignment. Each word needs to be related to the speech and video frames that occur while it is being spoken. For this task we use the Munich Alignment and Segmentation tool MAUS [8, 15] which uses phonological rules to find pronunciation variations and attempt to best match these for high quality speech alignment. While MAUS also provides the alignment of individual phonemes, we only use the word-level alignments in this work.

2.3.2 Face Detection and Extraction. We use OpenFace [3], an open source facial behavior analysis toolkit to detect and extract faces from videos. OpenFace uses the MTCNN [17] model to detect faces in the videos. Beyond the normalized face bitmaps, OpenFace produces a large number of features that estimate pose, gaze via 2D and 3D landmarks and facial action units [2] that attempt to directly model facial movements. We make use of these features in our models and analysis. During the extraction process there were several instances where OpenFace was unable to detect any face in the video clip (due to absence of face in the video, partial or complete occlusion, etc.). This finally leaves us with 6225 instances which we use to train and evaluate our models.

2.3.3 Audio feature extraction. We use MFCC features as a compact model of speech properties every 10 ms. (We experimented with Δ-features, however these did not yield improvements.)

3 MODELS

We first describe our model components: the video and audio encoder and the attention layer. We then describe how we use one or more of these components to create uni-modal or multi-modal models.

3.1 Model Components

The video encoder consists of a convolutional and a recurrent module. The convolutional module (CNN) takes as input an image sequence $\boldsymbol{v}$ and extracts image features (f_t) at each time step. These are fed to the (bidirectional GRU) recurrent module, which helps in capturing temporal dependencies. As CNN we use an Inception Resnet model pretrained on the CASIA-WebFace dataset [16]. The RNN takes as input the image features generated by the ConvNet to produce fixed dimensional output representation for each time step.

[1] The corpus includes a wide variety of pose, lighting, perspective, and artifacts from fast motion in DVD-encoded video.

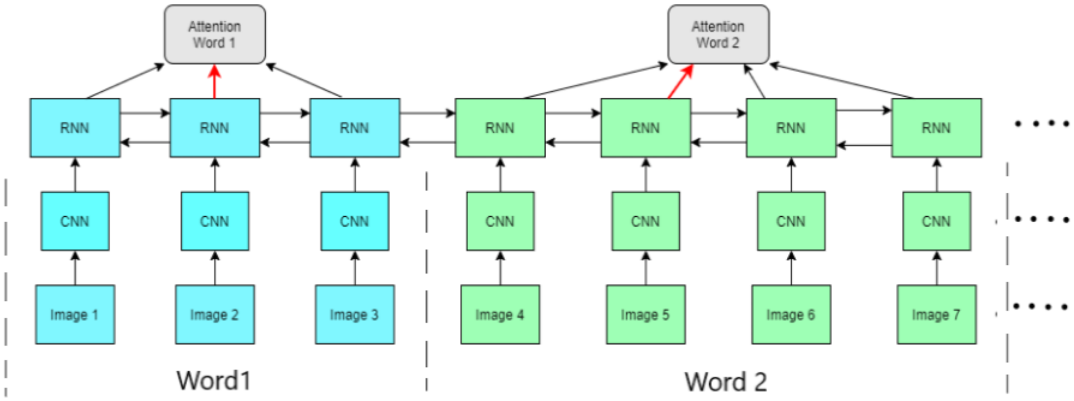

Figure 2: We use a mixture of hard attention (based on word alignments) and soft attention (using q/k/v systematics) to determine the representations to be passed into the decision layer for on/off-screen classification. (Not shown: audio representations which are handled similarly.)

The audio encoder consists of only a recurrent module which takes an audio sequence ***a***, composed of MFCC features as input and produces an output representation (o_t^a) for each time step.

The attention mechanism uses a mixture of soft and hard attention, allowing the model to identify the relevant RNN states (similar to [1], [9]). We make use of the time alignment of each word to identify frames relevant to that word. Word-level aggregation from frame-level features then uses the hidden state of the middle frame of the word as the key to the query vector consisting of all hidden states spanning the word (Fig. 2). This results in word level features. We believe that this mechanism would help to identify the frames which are relevant for the classification of that word and also aid in identifying relevant frames when multiple faces are present.

We also use a joint attention mechanism to combine the aggregated word level visual and audio features. For a particular word, the word level feature from the video encoder acts as the key and the aggregated word level features of all the words from the audio encoder acts as the query. This results in audio aware visual feature vectors.

The feature vectors for all the words, obtained from the joint attention mechanism are then passed through another bi-GRU layer, followed by a MLP with a softmax to generate the final class (on/off-screen) for each word in the utterance.

3.2 Implemented Model Configurations

3.2.1 Baseline using OpenFace. This model uses the 711-dimensional feature vector that OpenFace generates for each image as input to the RNN instead of a video encoding from images. The rationale for this baseline is that we expect the relatively low dimensional input to be suitable for our data-sparse setup. Also, OpenFace's action units are designed to be suitable for our task.

3.2.2 Video-only Model. This model uses the video encoder and the attention mechanism to produce features that are then passed into the decision layer as described above. We speculate that it might be capable of picking up more subtle details in facial images than what could be found by the action units.

3.2.3 Audiovisual Model. The audiovisual model builds on the intuition that it may help to relate the speech to the lip movements visible on screen in order to identify if the visible lips are those speaking or not. In this model, we use two encoders, one for audio and one for video, and follow this with a joint attention mechanism over both modalities to yield audiovisual representations per word to be passed to the decision-making component.

4 EXPERIMENTS

Our models are implemented in PyTorch [12]. The CNN module takes as input a 160×160 RGB image. Four 13-dimensional MFCC feature vectors are concatenated to form a 52-dimensional audio feature vector which acts as input to the audio encoder. The recurrent module in all our experiments consists of 50-dimensional bidirectional GRU units. We train the network using cross-entropy loss and the Adam optimizer [7] with an initial learning rate of .0005 which decreases exponentially with a step size of .9. Dropout is set to .5 for all layers but the attention, where the dropout is .3. We use 5-fold cross validation and results shown are medians of 3 runs to account for random initialization for a total of 15 result points per condition.

5 RESULTS

5.1 Word-level Classification

Table 1 shows the results for the model configurations described above. We use area under the Receiver Operating Characteristic curve (auROC) as the measure of performance (as a generalization of F-measure and as in [13]).

Table 1: Median (std) area-under ROC for all configurations

Model	auROC
OpenFace Baseline	.65 ± .016
Video-only Model	.71 ± .018
Audiovisual Model	.73 ± .016

Table 2: Face detection rate for the utterances in our corpus.

Utterances	Face detected
all	89.2 %
fully off-screen	68.5 %
fully on-screen	92.6 %
at least partially on-screen	93.5 %

Firstly, we observe that the baseline that directly uses OpenFace features has the lowest score. This indicates that these features insufficiently capture the facial movements necessary to detect stretches requiring lip-sync (or that facial action units are unreliable for real-life video). This confirms our initial intuition that automatic facial behaviour analysis tools are not sufficient for successfully detecting a speaking face on screen for the purpose of applying synchrony constraints and therefore there is ample space for developing customised solutions for the task. In fact, the performance is significantly higher (t-test, $p < .002$) for our proposed Video-only model.

Secondly, the Audiovisual model outperforms the Video-only model significantly (t-test, $p < .005$), which suggests that it can learn the correspondence between audio and the mouth movements (or absence thereof for off-screen words), leading to better classification. Audio is thus helpful to distinguish arbitrary mouth movements of a listener to the speaker's mimicry.

5.2 Face Detection Performance

Our on/off-screen classification relies on the performance of the face detector used (and we ignored cases where no face was found at all in Table 1). We measured for each utterance whether a face was detected in at least one of the corresponding video frames and report the results in Table 2. As can be seen in that table, about 90 % of the utterances co-occur with a face detection. As we have not manually annotated all frames for faces (but merely marked whether speech was on- or off-screen), it is hard to estimate the correctness of the face detector. However, we assert that a face should be visible in utterances that are fully or partially on-screen.

The face detector misses a face in utterances in which a face *should* be visible at least part of the time (according to the annotation) in 6.5 % of the cases. Based on a manual analysis, we believe that the visibility of a part of the face only, low brightness of the scene, small size of the speaking face are the main reasons for the failure of face detection. At the same time, utterances that are spoken fully off-screen still have a substantial rate of face detection. This matches our observations that non-speaking faces are often visible while the speech comes from a speaker off-screen (cmp. Figure 1, left side). This again shows the importance of a specialised solution for the task of detecting stretches requiring lip-sync. Overall, we find that our system's performance may be impeded by the two-stage approach and the errors of the face detection stage. In particular if faces are not detected for a certain duration of on-screen speech, the corresponding words will be classified by the model as off-screen (majority class for such cases).

Table 3: Accuracy (mean and std dev) when aggregating word-level classifications to the utterance level

Class	Accuracy (audiovisual)	Baseline (random)
on-screen	.55 ± .08	0.18
off-screen	.50 ± .07	0.02
mixed with ≤ 3 word errors	.38 ± .01	.30

5.3 Utterance-level Analysis

We aggregate the word-level predictions made by the model to obtain utterance-level predictions (for the audiovisual model) in Table 3 and contrast this with a random baseline. If decisions differ between words in an utterance, we count these as 'mixed' and we also compute the accuracy for utterances where the 'mixed' decision matches the corpus with ≤ 3 word errors. This analysis is useful to understand the sequential behaviour of the classifier, as well as to identify which types of utterances are particularly challenging.

Table 3 shows that the per-word classification works without flaws more often for utterances that are completely on-screen than for off-screen utterances. By further manual analysis, we find that for the on-screen case, the performance is hurt by the detector used to extract faces from the video, as described in the previous subsection. We conclude that better face detection (tuned to the characteristics of the data) would be beneficial. Further, we have an imbalanced and small data set where on- to off-screen words are in the ratio (2.5 : 1). While we accounted for this during training by over-sampling the smallest class, there are still very few off-screen examples available to learn from. Moreover, during the face detection stage we exclude many off-screen utterances as no face was visible on screen for these instances. We believe that this imbalance hurts the performance of off-screen classification.

For the mixed class, the model often fails to accurately detect the boundary between on- and off-screen words (but still outperforms the random baseline). We also experienced the difficulty of precisely identifying the boundary between on- and off-screen speech while annotating the data.

6 CONCLUSIONS

We have presented the task of determining for every word that is spoken in a video whether the speaking face is visible on screen or not. This differentiation is useful for fine-grained control of automatic dubbing, as it allows for adding synchrony constraints to generate translations respecting the norms of visual phonetics [6] only for those stretches of speech where lips are visible on screen. We find that we can approach this task achieving reasonably good performance, in particular when taking into account both audio and video in a multi-modal classifier, despite the higher number of parameters to be trained. We believe that both the classifier as well as the gold-standard annotations will be an important component to accelerate the design and implementation of machine translation solutions for dubbing.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Y. Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. *Proceedings of ICLR 2015.*

[2] Tadas Baltrušaitis, Marwa Mahmoud, and Peter Robinson. 2015. Cross-dataset learning and person-specific normalisation for automatic action unit detection. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, Vol. 6. IEEE, 1–6.

[3] Tadas Baltrusaitis, Amir Zadeh, Yao Chong Lim, and Louis-Philippe Morency. 2018. OpenFace 2.0: Facial Behavior Analysis Toolkit. In *Automatic Face & Gesture Recognition (FG 2018), 2018 13th IEEE International Conference on.* IEEE, 59–66.

[4] Frederic Chaume-Varela. 2006. Dubbing. In *Encyclopedia of Language & Linguistics (Second Edition)*, Keith Brown (Ed.). Elsevier, Oxford, 6 – 9. https://doi.org/10.1016/B0-08-044854-2/00471-5

[5] Marcello Federico, Robert Enyedi, Roberto Barra-Chicote, Ritwik Giri, Umut Isik, Arvindh Krishnaswamy, and Hassan Sawaf. 2020. From Speech-to-Speech Translation to Automatic Dubbing. In *Proceedings of the 17th International Conference on Spoken Language Translation.* Association for Computational Linguistics, Online, 257–264. https://www.aclweb.org/anthology/2020.iwslt-1.31

[6] I. Fodor. 1976. *Film Dubbing: Phonetic Semiotic, Esthetic & Psychological Aspects.* Buske Helmut Verlag Gmb.

[7] D. P. Kingma and J. Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015,*. http://arxiv.org/abs/1412.6980

[8] Thomas Kisler, Florian Schiel, and Han Sloetjes. 2012. Signal processing via web services: the use case WebMAUS. In *Digital Humanities Conference 2012.*

[9] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. *CoRR* abs/1508.04025 (2015). arXiv:1508.04025 http://arxiv.org/abs/1508.04025

[10] Alp Öktem, Mireia Farrús, and Antonio Bonafonte. 2018. Bilingual Prosodic Dataset Compilation for Spoken Language Translation. In *Proceedings of Iber-SPEECH 2018* (Barcelona, Spain, 21-23 November 2018). 20–24. https://www.isca-speech.org/archive/IberSPEECH_2018/pdfs/IberS18_P1-1_Oktem.pdf

[11] Pilar Orero. 2004. *Topics in audiovisual translation.* Vol. 56. John Benjamins Publishing.

[12] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[13] Joseph Roth, Sourish Chaudhuri, Ondrej Klejch, Radhika Marvin, Andrew C. Gallagher, Liat Kaver, Sharadh Ramaswamy, Arkadiusz Stopczynski, Cordelia Schmid, Zhonghua Xi, and Caroline Pantofaru. 2019. AVA-ActiveSpeaker: An Audio-Visual Dataset for Active Speaker Detection. *CoRR* abs/1901.01342 (2019). arXiv:1901.01342 http://arxiv.org/abs/1901.01342

[14] Ashutosh Saboo and Timo Baumann. 2019. Integration of Dubbing Constraints into Machine Translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers).* Association for Computational Linguistics, Florence, Italy, 94–101. https://doi.org/10.18653/v1/W19-5210

[15] Florian Schiel. 2004. MAUS goes iterative. In *Proceedings of the LREC.*

[16] Dong Yi, Zhen Lei, S. Liao, and S. Li. 2014. Learning Face Representation from Scratch. *ArXiv* abs/1411.7923 (2014).

[17] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. 2016. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Processing Letters* 23, 10 (2016), 1499–1503.