# CapCap: An Output-Agreement Game for Video Captioning

*Hernisa Kacorri*[1], *Kaoru Shinkawa*[2], *Shin Saito*[2]

[1]The Graduate Center, City University of New York, NY – USA
[2]IBM Research - Tokyo, IBM Japan, Tokyo – Japan
hkacorri@gradcenter.cuny.edu, {kaoruma, shinsa}@jp.ibm.com

## Abstract

CapCap is an output-agreement game that challenges players' listening and speaking skills. Players submit their transcriptions for short video segments against a countdown timer, in one of three pre-specified modes, to score points and support their team. Adding entertainment value, the game channels input toward captioning videos without monetary rewards. It deploys a novel human computation algorithm, which collects input from a crowd of non-experts, sequentially and in parallel, until a completion criterion is met. Rather than monetary incentive, CapCap uses motivational mechanisms like indirect feedback, mix of player skills, and community identification. Preliminary results from a field trial with mostly non-native English speakers improved the WER of English captions over ASR output.
**Index Terms**: gamification, captioning, transcription, crowdsourcing, human-computer interaction

## 1. Introduction

Video captions and audio transcriptions make audio content accessible for people who are deaf or hard-of-hearing, support indexing and summarization, and can help in vocabulary acquisition for second language learners [1]. Although Automatic Speech Recognition (ASR) has advanced significantly, it still requires human transcribers for accurate captioning. There are three challenges arising in the current captioning pipeline. First, there is a shortage of trained transcribers, e.g stenographers (real-time) and language-competent typists (pre-recorded media). Moreover, subject-matter knowledge may be crucial. This has led researchers to investigate high-accuracy captions by multiple non-experts in crowd-sourced platforms, e.g. [2]. Second, better user interfaces are required to speed up manual processes requiring transcribers' sustained attention, both for reading and listening. Some proposed solutions involve segmentation by pause detection [3] and efficient editing interfaces [4]. Finally, scaling to caption growing media collections requires cheaper crowd-captioning services e.g. through non-monetary incentives. Games with a purpose [5] were proposed to attract crowd engagement by adding entertainment value. Unfortunately, captioning has limited entertainment value; user input is closely bound to the media and there is little room for creativity. This means other motivational approaches are needed.

In this paper we describe CapCap, an output-agreement game [16] that provides video captions while challenging players' English listening and speaking skills. The players can be a crowd of non-experts including second language learners, as in Duolingo [6]. CapCap deploys a new human computation algorithm that combines ASR output, players' input (through sequential and in-parallel processes), and task-completion criteria. The game's motivational mechanisms are based on a Crowdsourcing Motivation Model proposed in [7].

A few other systems have adopted crowdsourcing for transcription. Legion Scribe [2] requests in-parallel real-time audio captions from MTurk [8] and merges them through majority voting. It incorporates some game-like elements. Transcription Game [9], a single-player game with monetary rewards, iterates over dual paths to converge to a final audio transcription. Synote [10] users edit ASR errors in parallel and the final result is selected through matching-and-voting algorithms. It incorporates some game-like elements. CastingWords [11] uses MTurk to transcribe audio, correct, and score transcripts. Voice Scatter [12] uses MTurk and selects transcriptions with majority vote on exact agreement. PodCastle [4] users transcribe audio and video either by selecting from a list of candidates or by typing the correct text. [13] investigates incremental redundancy and ASR as a worker in MTurk, where most frequent transcription is obtained from in-parallel input. CapCap differs from these systems in several ways. It is a game that supports both two- and single-player modes as part of an end-to-end video captioning system. Its design incorporates various player skills, such as listening, speaking, reading, and writing. Input is collected from players both iteratively and in parallel. This is in contrast to the in-parallel only, iterative only, or two iterative-paths approaches of the other systems. Differences in these choices can impact both accuracy and efficiency (discussed in [9] and [14]).

## 2. Game Play

CapCap is based on the fundamental input-output behavior of the ESP game for image labeling [15] as adapted for video captioning. Solving a different computational problem than ESP led CapCap game play and mechanisms to vary as well.

**Initial Setup.** During registration, players select nicknames, self-rate listening and speaking skills, and join teams. A game begins by randomly matching two players.

**Rules.** In each round, two players, given the same short video segment, must guess what the other player reports hearing (Fig. 1a). A round mode can be:

- TYPE: The player types a guess.
- FIX: The player is given a suggestion, perhaps helpful, and attempts to correct it where appropriate.
- SPEAK: The player imitates the spoken phrase, which is then transcribed using ASR and submitted without edits.

Each game session has 6 rounds, illustrated in Fig. 1b. Players cannot communicate directly or see the other players' outputs during the game. The game prompts a player to try and guess the other player's output. Since the players do not know each other's identity or output, the results tend to be close to the correct captions. Each round has a countdown clock. Players can skip or submit a guess before the clock expires. Alternatively, the current text is automatically submitted.

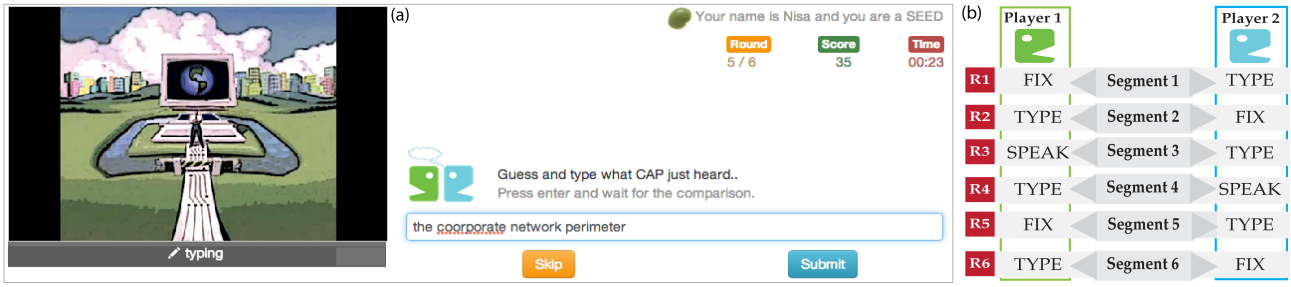


Figure 1: *CapCap (a) TYPE round instance and (b) round-type pairs in a game session.*

**Winning Condition.** In any mode, players submit text for their answers. Scores are calculated based on degree of agreement by counting matching words. Two exceptions apply: (i) in FIX rounds, if a player doesn't modify the suggested text, only half the points are awarded and, (ii) in SPEAK rounds, each agreement scores double.

Scores are tallied at the game end and highest scores are updated for players. Scores are also aggregated over teams and define a user's contribution and rank within their team. Our contribution ranking is similar to Peekaboom [16] with a naming convention: Seed, Sprout, Leaf, Stem, Branch, and Tree.

## 3. Game Mechanisms for Task Completion

The input media submitted to CapCap is automatically segmented by pause detection, which are placed in a pool to await transcription. The game begins by randomly matching two players and assigning six random video segments to the rounds (Fig. 1b). If players can't be matched, CapCap switches to a single-player mode, but simulates a multi-player game by matching rounds from two single player games.

### 3.1. When Is a Video Segment "Done"?

CapCap deploys a human computation algorithm (Alg. 1) to detect if a segment transcription is complete and nudge players' inputs closer to the truth. ASR transcription of a segment is used to initialize suggested text for the first FIX round. A historical list (H) of transcriptions from ASR and all games is maintained (Line 2). After each round, transcriptions submitted by each player are appended to list H (Lines 7, 10) and the completion criteria are checked (Lines 11-18). If satisfied, the segment is removed from the pool (and all future games), and the final caption is calculated by merging all transcriptions in H (Line 20). Otherwise, it remains in the pool with its suggested text updated for the next FIX round (Line 13). H[-1] and H[-2] denote the final and penultimate entries of H, respectively. A segment is successfully transcribed either when two consecutive FIX rounds pass without any edits, or the segment has been seen in a pre-set number of rounds.

**Algorithm 1** Video Segment Crowd-Captioning

```
1:  function COMPLETE(segment)
2:      suggestion = asr, H = [asr]
3:      maybe_done = false , done = false , played = 1
4:      while (not done) do
5:          if played is odd then
6:              typed = TYPE_round(segment)
7:              APPEND(H, typed)
8:          else
9:              fixed = FIX_round(segment, suggestion)
10:             APPEND(H, fixed)
11:             if fixed != suggestion then
12:                 maybe_done = false
13:                 suggestion = MERGE([H[-1], H[-2]])
14:             else
15:                 if maybe_done != true then
16:                     maybe_done = true
17:                 else
18:                     done = true
19:         played += 1
20:     return MERGE(H)
```

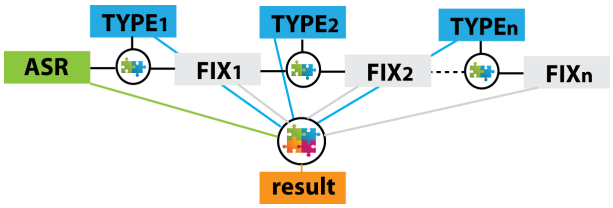


Figure 2: *Video segment crowd-captioning process.*

The game balances exploitation and exploration to constrain and search the space of candidate player transcriptions. In FIX rounds, players exploit a constrained space by sequentially improving suggested transcriptions. In TYPE rounds, players explore the search space through parallel contributions. When completion criteria are met, the final result is calculated using all candidates (Fig. 2). The merging process (Lines 13, 20) is similar to ROVER [17] (see Implementation section).

We assume FIX transcriptions have fewer typos and are more complete, while TYPE transcriptions are not misled by errors from the ASR or other players. This helps explain some of the counterintuitive decisions of Algorithm 1:

**Why not take the last suggested text as the final result?** The game is structured to refine transcriptions per FIX round. However, this is not guaranteed. An almost correct suggestion may prevent players from modifying the transcription.

**Why isn't the suggestion always updated with the newest FIX transcription or by combining all previous transcriptions?** While a FIX round inherently carries information about previous rounds, it isn't guaranteed to improve results, so later players may be misled, delaying convergence. [18] suggests that users make fewer changes in captions sufficiently close to the ground truth. Merging the outputs of latest FIX and TYPE per each suggestion update is more efficient than multiple sequence alignment on all submitted transcriptions.

**Why isn't SPEAK round considered for completion?** As proposed in [19], echoing speech for better ASR results requires a trained speaker and settings not guaranteed by CapCap. However, this round can be used to generate self-labeled speech data.

# 4. Game Motivation Mechanisms

For crowd-captioning, we have to attract and maintain a crowd of players and ensure the caption quality is acceptable. [7] lists several motivational factors correlated with weekly time spent on MTurk. While designing CapCap, we focused on their strongest factors as potential incentives.

*Human Capital Advancement:* CapCap challenges players' listening, reading, typing, and speaking skills with time constraints on a second language or an extended vocabulary in the player's native language. We consider this factor as a potential incentive given the inherent nature of listening and trying to understand audio in a target language [20], [21], and [22]. However, our current work was not assessed for educational results.

*Indirect Feedback:* Players' input are compared against each other and their disagreements are displayed. Since segments are chosen at random, a frequent player may see the same segments with new suggestions providing further feedback.

*Skill Variety:* CapCap requires multiple skills (listen, read, type, speak) in each game session, where segments are drawn randomly from various media, topics, vocabulary, and speakers.

*Community Identification:* Team membership enhances community identification. E.g., players may be motivated to contribute by supporting the accessibility goals of a community. CapCap displays both daily team rankings based on aggregated scores and internal team information for the players' ranks.

*Signaling:* Speaking and listening skills can be important for employment. A user may seek to enhance their ranking in CapCap to gain reputation for language skills.

# 5. Implementation

CapCap is a Web-based game. Initial ASR results and the real-time transcriptions of the players' recorded voices were obtained using the IBM Attila speech recognition toolkit [23]. The game is part of an end-to-end system [3], where requesters upload their media. Here are some details on the tuning parameters for the game mechanisms and the gameplay characteristics:

*Score Calculation:* Transcriptions are aligned using a word-level Levenshtein distance [24]. Exact string comparison is used for each word pair to provide feedback on spelling errors (Table 1). The number of correctly aligned words is multiplied by a constant: 0.5 for no edits in FIX, 2 for SPEAK, or 1 otherwise.

*Suggestion Update:* The last two transcriptions in list H (Alg. 1), either (ASR, TYPE) or (TYPE, FIX) are aligned using a word-level Levenshtein match that allows for substitutions (ins/del 1 and sub 3). Two words (converted to lowercase) are the same when: (i) the direct string comparisons match, (ii) they are homophones, or (iii) their character-level Levenshtein distance (ins/del 1 and sub 2) is at most 25% of the unaligned distance. The merging phase uses heuristics to pick one word, such as for homophones, the TYPE word is favored over both the ASR and FIX round e.g. 'disk' over 'disc' (Table 1).

*Final Results:* All partial transcriptions in list H (Alg. 1), including the initial ASR result, are aligned using Multiple Sequence Alignment [25] with un-weighted A* search for efficient and optimal alignment. Our implementation uses pairwise word-level alignment and majority vote for merging as in [26].

*Voice sampling:* Player's voice is sampled at 11.025 KHz.

Table 1: *Scoring and suggestion update examples.*

| | | | | | | |
|---|---|---|---|---|---|---|
| Player 1 | the | big | benefit | after | several | quarters |
| Player 2 | a | big | benefit | | several | quarter |
| Score | 0 | 1 | 1 | 0 | 1 | 0 |
| | | | | | | |
| ASR | it | must | | now | consider | disc |
| TYPE | IT | must | not | | consder | disk |
| SUGG. | IT | must | not | now | consider | disk |

While a higher sampling rate may obtain better real-time ASR it causes longer delays that can negatively affect the game play.

*Round Duration:* [3] indicated that it takes non-experts about 8 times the length of a video to add captions. Thus our rounds timeout at the length of each segment multiplied by 9.

*Video Display:* Each video segment plays in a loop, stops when the player is editing, and resumes at a slightly earlier point. After a transcription is submitted, the segment plays once more while CapCap shows the agreement result and score.

*Teams:* The game was pre-populated with 11 teams based on geographical locations. Teams were reordered daily based on the cumulative scores of their members. Players can access detailed information about their team, e.g. the number of players at each rank, but can only see total scores for other teams.

*Feedback:* Players can provide their feedback (five-star rating and comments) at the end of each game.

# 6. Evaluation

We announced CapCap to 250 full-time employees of a corporation as a video captioning game that challenges their English speaking and listening skills. 105 people registered in a 3-week period and 66 played at least an entire 6-round game. 16 players self-reported English literacy skills on a 1-5 scale, with a mean of 2.75 (speaking) and 2.8 (listening). The reported native languages were: Japanese (81), Chinese (3), Hindi (3), Portuguese (2), Greek (1), and English (15). We attracted a small, yet diverse, group with a variety of English literacy skills.

We pre-populated CapCap with 25 selected videos (1-5 minutes long) from a corporate Media Library, with expert-transcribers base-truth captions. To retain content-independent motivation, selected videos had limited entertainment value. They included a total of 30 speakers; 10 narratives, 5 dialogs, and 7 (noisy) conference recordings. 41 minutes of video content were segmented into 637 short clips (2-10 seconds each).

CapCap evaluation focused on transcription accuracy, game-mechanism efficiency, and early results on playability.

## 6.1. Accuracy of Collected Data

During the 3-week period, 60 video segments were transcribed while additional 577 were partly transcribed. To evaluate their accuracy, we compared the Word Error Rate (WER) of the results produced by CapCap to the results from (i) ASR, (ii) first TYPE rounds, and (iii) first FIX rounds; [26] observed strong agreement of WER with human participants for evaluating transcription accuracy. WER were calculated in terms of exact match with the experts? transcriptions, ignoring whitespaces, capitalization, and punctuation.

Overall WER for completed segments improved by 8.4% with CapCap over the ASR's 4%. Figure 3 shows distributions of WER for completed segments as boxplots with 1.5 interquartile range (IQR). For comparison, medians are labeled above

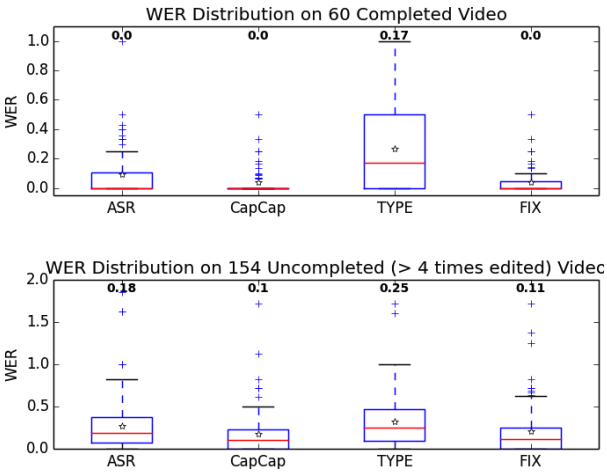


Figure 3: *WER distributions for CapCap results.*

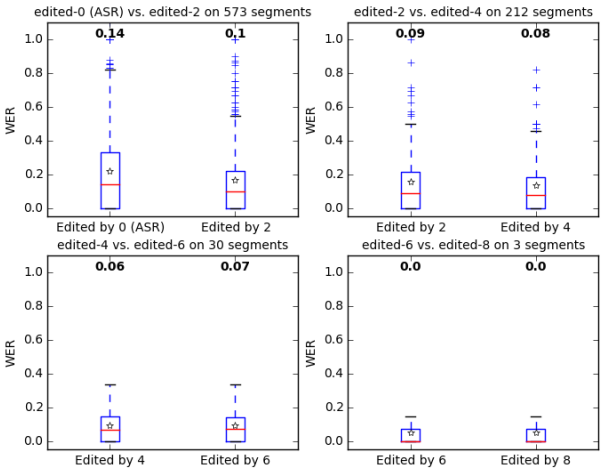


Figure 4: *WER between rounds.*

each plot and means are denoted with a star. CapCap achieves a much smaller variance. About 80% of segments converged in 4 rounds (the minimum number of rounds for completion in Alg. 1). This suggests that ASR results for these segments already had high accuracy and the suggested texts were not further improved by the players during FIX rounds. In addition, CapCap improved upon transcriptions produced in the first TYPE round and improved slightly on transcriptions submitted during the first FIX rounds. We also observed that "misled-by-ASR players" were often responsible for errors in CapCap results, e.g. mapping "blocks" to the accurate word "blogs". However, in many cases CapCap overcame some ASR errors such as "w.'s free" to "w3" and "meat market" to "midmarket".

The sample of 60 completed segments is quite small, and includes many segments with high ASR accuracy. Thus, we calculated WER for remaining segments in the game, which had not met completion criteria but were edited by at least 4 players. These totaled to 154 segments, and overall WER dropped to 17.3% (CapCap) from 25.6% (ASR). Fewer than 5 players edited about 80% of such segments. Figure 3 shows, for uncompleted transcriptions, that CapCap WER achieved a lower variance than ASR, first TYPE rounds, or first FIX rounds.

### 6.2. Game Design Efficiency

We investigated game efficiency in terms of WER relative to number of submissions for a segment transcription. Given the poor real-time recognition rate in SPEAK rounds, we considered only transcriptions obtained by TYPE or FIX rounds. All 637 segments (completed and uncompleted) played at least once in the game were included. Table 2 shows aggregated results of videos grouped by transaction history. The statistics represent partial results at each grouping through the transaction history, where T and F stand for TYPE and FIX rounds, respectively.

Table 2: *Partial results grouped by transaction history.*

| Transaction History | # Segments | #Words | WER |
|---|---|---|---|
| 0 (ASR) | 637 | 7,227 | 20.3% |
| 2 (ASR,T1,F1) | 573 | 6,501 | 16% |
| 4 (ASR,T1,F1,T2,F2) | 212 | 2,336 | 14% |
| 6 (ASR,T1,F1,...,T3,F3) | 30 | 330 | 10% |
| 8 (ASR,T1,F1,...,T4,F4) | 3 | 31 | 6.5% |

Given that fewer segments survive at each additional round, we draw WER distribution between rounds based on common segments (Fig. 4). There were a total of 573 segments edited by 2 players with median WER lower than that of ASR. The variance of the WER also tends to be lower, as shown by smaller whiskers and IQRs. However, the boxplot reveals a number of outliers. When we compare 212 segments edited by 2 players versus 4, we observe a decrease in WER median and variance for 4 players. In addition, there are fewer outliers with more players. Above that point, we only have small samples, with 30 segments edited by 6 players and 3 segments by 8. However, the general trend seems to be decreasing with converging medians and lower variances for the WER.

### 6.3. Playability

A total of 66 people played the game (17 people joined the first week, 15 the second, and 34 the third), generating 2191 total transcripts for 713 different segments through all types of rounds. To gain some perspective on game engagement, we observed that 10% of the users spent more than 46 minutes playing and 90% more than 3 minutes, which is about the duration of 2 game sessions. Over 21 days, 10% of users played more than 4 days, with at most 20 games played in a day. We received 45 comments from 31 players. Some of the positive comments include: "fun! can't stop it!", "3 times is a charm! highest score yet!", and "I did better the second time around :)".

## 7. Conclusions

We proposed CapCap, a system that adapts gamification to harvest crowdsourcing for video captions. CapCap addresses a number of challenges in captioning systems. First, it enables captioning by a crowd of non-experts who can contribute through team efforts. Second, it incorporates a user-friendly interface that assigns short video segments to micro-tasks, which are rotated to maintain users' attention. Third, CapCap offers video captioning without monetary rewards by activating additional motivational factors. CapCap evaluation yielded positive results in confirmation with the proposed approach. Future work will investigate the relationship between the motivational factors in crowdsourced captioning and the transcription accuracy, as well as the convergence rate to accurate results.

# 8. References


[1] J. Krajka, "Audiovisual translation in LSP-A case for using captioning in teaching languages for specific purposes," *Scripta Manent*, vol. 8, no. 1, pp. 2–14, 2013.

[2] W. Lasecki, C. Miller, A. Sadilek, A. Abumoussa, D. Borrello, R. Kushalnagar, and J. Bigham, "Real-time captioning by groups of non-experts," *Proceedings of the 25th annual ACM symposium on User Interface Software and Technology*, pp. 23–34, ACM, 2012.

[3] A. Sobhi, R. Nagatsuma, and T. Saitoh, "Collaborative caption editing system – enhancing the quality of a captioning and editing system," *Proceedings of the Annual International Technology and Persons with Disabilities Conference (CSUN)*, 2012.

[4] M. Goto and J. Ogata, "PodCastle: Recent advances of a spoken document retrieval service improved by anonymous user contributions," *Interspeech 2011 – 14th Annual Conference of the International Speech Communication Association Proceedings*, pp. 3073–3076, 2011.

[5] L. von Ahn and L. Dabbish, "Designing games with a purpose," *Communications of the ACM*, vol. 51, no. 8, pp. 58–67, 2008.

[6] Duolingo – www.duolingo.com

[7] N. Kaufmann, T. Schulze, and D. Veit, "More than fun and money. Worker motivation in crowdsourcing-A study on Mechanical Turk," *AMCIS*, 2011.

[8] Amazon Mechanical Turk (MTurk) – www.mturk.com

[9] B. Liem, H. Zhang, and Y. Chen, "An Iterative dual pathway structure for speech-to-text transcription," *Human Computation*, 2011.

[10] M. Wald, "Concurrent collaborative captioning," *Proceedings of SERP – The 2013 International Conference on Software Engineering Research and Practice*, 2013.

[11] CastingWords – castingwords.com

[12] A. Gruenstein, I. McGraw, and A. Sutherland, "A self-transcribing speech corpus: collecting continuous speech with an online educational game," *Proceedings of SLaTE Workshop*, 2009.

[13] J. D. Williams, T. Alonso, B. Hollister, and J. Wilpon, "Crowdsourcing for difficult transcription of speech," *In Automatic Speech Recognition and Understanding (ASRU), IEEE Workshop*, pp. 535–540, 2011.

[14] G. Little, L. Chilton, M. Goldman, and R. Miller, "Exploring iterative and parallel human computation processes," *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pp. 68–76, 2010.

[15] L. von Ahn and L. Dabbish, "Labeling images with a computer game," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 319–326, ACM, 2004.

[16] L. von Ahn, R. Liu, and M. Blum, "Peekaboom: a game for locating objects in images," *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 55–64, ACM, 2006.

[17] J. Fiscus, "A post-processing system to yield reduced error rates: recognizer output voting error reduction (ROVER)," *Proceedings of IEEE ASRU Workshop*,1997.

[18] R. Nagatsuma, K. Fukuda, Y. Yaginuma, and Y. Hirose, "Effective captioning method by using crowd-sourcing approach," *Proceedings of SIG-WIT Well-being Information Technology*, 2012.

[19] S. Miyoshi, H. Kuroki, S. Kawano, M. Shirasawa, Y. Ishihara, and M. Kobayashi, "Support technique for real-time captionist to use speech recognition software," *Computers Helping People with Special Needs*, pp. 647–650, Springer Berlin Heidelberg, 2008.

[20] R. Schmidt, "Consciousness and foreign language learning: A tutorial on the role of attention and awareness in learning," *Attention and awareness in foreign language learning*, pp. 1–63, 1995.

[21] J. Field, "Skills and strategies: Towards a new methodology for listening," *ELT journal*, vol. 52, no. 2, pp. 110–118, 1998.

[22] J. Truscott, "Noticing in second language acquisition: A critical review,", *Second Language Research*, vol. 14, no. 2, pp. 103–135, 1998.

[23] H. Soltau, G. Saon, and B. Kingsbury, "The IBM Attila speech recognition toolkit," *Spoken Language Technology Workshop (SLT)*, pp. 97–102, 2010, IEEE.

[24] D. Sankoff and J. B. Kruskal, "Time warps, string edits, and macromolecules: the theory and practice of sequence comparison," *Reading: Addison-Wesley Publication*, 1983, edited by Sankoff, David; Kruskal, Joseph B., 1.

[25] R. C. Edgar and S. Batzoglou, "Multiple sequence alignment," *Current opinion in structural biology*, vol. 16, no. 3, pp. 368–373, 2006.

[26] I. Naim, D. Gildea, W. Lasecki, and J. P. Bigham, "Text Alignment for Real-Time Crowd Captioning," *Proceedings of NAACL-HLT*, pp. 201–210, 2013.