

**AutoSSD: A System for Automated Detection of
Similar Speech Responses in Language Tests**

Michael Fauss, Jiangang Hao, Chen Li, Molly Palmer, and Ikkyu Choi
Educational Testing Service, Princeton, NJ

Author Note

The authors would like thank the ETS Enterprise Test Security Initiative and ETS Research Allocation, who funded this work, and the many ETS colleagues who contributed by reviewing outputs, preparing audio files, and running AutoSSD. In particular, we would like to thank Jakub Novak, Wenju Cui, and Michael Angelo.

Conflict of Interest Disclosure

The authors have no conflicts of interest to declare.

Abstract

Evaluating spoken language proficiency stands as a pivotal component in language assessment. A well-known method for cheating in spoken language tests involves reciting pre-scripted responses. This paper describes a system called the Automated Speech Similarity Detector (AutoSSD). AutoSSD leverages state-of-the-art AI transcription to convert spoken responses into texts, then calculates various metrics of verbatim similarity between them, flags pairs whose similarity levels exceed set thresholds, and provides an interactive interface for a subsequent expert review. Our system has proven effective in practical operation, and our findings can help other spoken language assessments bolster their test security measures.

Keywords: Language Assessment, Test Security, Speech Similarity Detection

AutoSSD: A System for Automated Detection of Similar Speech Responses in Language Tests

Introduction

The introduction of remote testing in the wake of the COVID-19 pandemic brought tremendous convenience to learners and test takers (St-Onge et al., 2022; Zheng et al., 2021) but also opened the doors for potential cheating (Bilen & Matros, 2021; Janke et al., 2021; Newton & Essex, 2023). One of the simplest yet most common ways of cheating is to reuse existing material (Noorbehbahani et al., 2022). Test takers can copy prepared responses to leaked or reused items (Y. Chen et al., 2022) or use generic templates that can fit a wide variety of prompts with minimal modifications. Moreover, in remotely administered tests, proctors rely on cameras (often a single one) with a limited field of view. These circumstances add new challenges in ensuring test security, thus motivating the use of AI-based, automated systems that can detect cheating attempts *post-factum*.

Many language tests include a speaking section that is prone to the above-mentioned type of cheating, namely, test takers reading out prepared or copied responses. The Automated Speech Similarity Detector (AutoSSD) was designed to detect these cheating attempts. Since similarity/plagiarism detection for written text is a much more mature area of research (Foltýnek et al., 2019; Gomaa & Fahmy, 2013), the underlying idea is to make the latter applicable to spoken responses by transcribing audio recordings. Transcribing spoken responses accurately and cost-efficiently has been a significant challenge for a long time. However, the swift advancement of AI technology in recent years has matured speech-to-text transcription, making the detection of similar spoken responses practical and feasible.

In a nutshell, AutoSSD works by calculating pairwise verbatim similarities of all responses submitted in a test administration. The exact similarity measures used for this purpose will be detailed in a later section. If the similarity of an essay pair exceeds a certain threshold, both test takers are flagged since they may have copied their responses

from each other or, more likely, from the same template. Using the same principle, AutoSSD also compares submitted essays to prompt text to flag test takers whose response is merely a repetition or reformulation of the question. All flagged response-response and response-prompt pairs are then forwarded to a human expert review to make the final decision whether the detected similarities constitute plagiarism.

AutoSSD is a continuation of a line of work on plagiarism detection for spoken responses. Similarities to the prompt questions were used by Yoon and Xie, 2014 to detect responses that are unfit for automated scoring. For plagiarism detection, Evanini and Wang, 2014; X. Wang, Evanini, Mulholland, et al., 2019; X. Wang et al., 2016 employed a text-similarity-based methodology conceptually similar to the one proposed in this paper. Based on these results, X. Wang, Evanini, Qian, and Zechner, 2019 explored the use of deep neural networks to learn similarity patterns between plagiarized responses. AutoSSD builds on these works, but has a stronger focus on applicability and scalability.

Before turning to the main sections, we offer two clarifying remarks regarding the purpose and context of this paper. First, it is important to note that this paper is primarily descriptive: We detail the architecture and working principles of AutoSSD and report its performance over a six-month period of use with real data. The paper is not intended to present the results of a rigorously designed and conducted empirical study, nor do we consider it a contribution to advancing the state-of-the-art in speech similarity detection methodology—AutoSSD is based on well-known ideas and techniques. Nevertheless, we believe that sharing our observations and experiences benefits the test security and speech processing communities. In contrast to many proposed systems that have only been demonstrated under lab conditions and on comparatively small benchmark datasets, AutoSSD was integrated into a live processing pipeline and has been extensively tested on large-scale real data. Therefore, we consider the AutoSSD architecture a useful starting point for practitioners or applied researchers who are faced with similar challenges.

Second, We would like to highlight that this paper provides insights into a

plagiarism system used in operation by a large testing organization. Typically, these systems are proprietary and opaque to both researchers and the test takers affected by them. With this paper, we aim to be as transparent as possible without revealing information that could potentially be used to compromise the security of our tests. Therefore, our description of the system and presentation of the results can lack details that are typically expected in a research report. We kindly ask the reader to keep this in mind.

The remainder of the paper is organized as follows: In System Architecture, the overall architecture of AutoSSD is detailed. In Transcription, the Automatic Speech Recognition (ASR) used by AutoSSD to transcribe the spoken responses is discussed. The similarity measures and thresholds used for detecting suspicious response-response and response-prompt pairs are described in Detector. Results and Findings summarizes our observations and findings from using AutoSSD in a large language test (ETS, 2025b), between November 2022 and March 2023. The paper concludes with a discussion of the strength and weaknesses of AutoSSD and an outlook on possible extensions and alternative approaches.

System Architecture

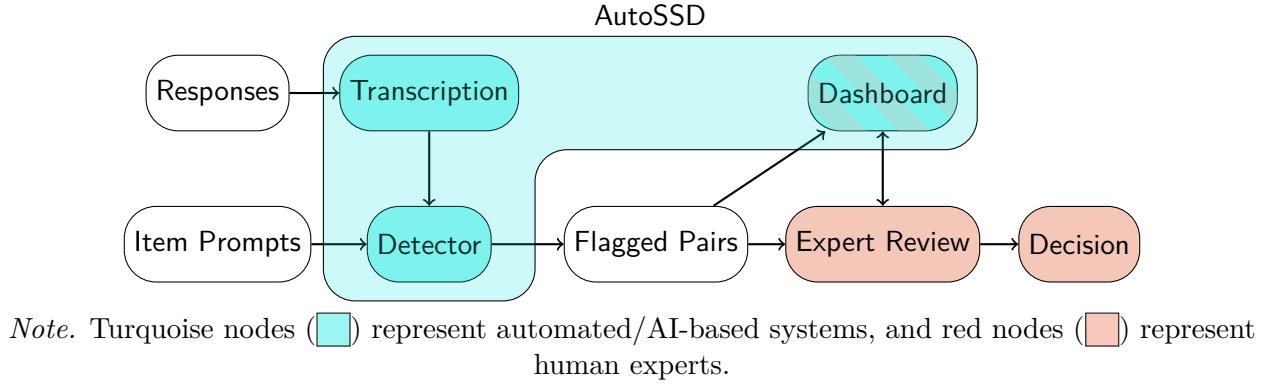
In this section, the architecture of AutoSSD is introduced and explained in broad strokes. The AutoSSD pipeline is depicted in Figure 1, with “Transcription” and “Detector” being the core blocks. These will be discussed in more detail in later sections.

The AutoSSD pipeline consists of the following six components:

- 1. Responses and Prompts:** The inputs to AutoSSD are audio recordings of the test takers’ speech responses and the corresponding item prompts. Depending on the item type, the latter can range from a short stem for independent speaking tasks to a combination of stems, reading passages, and lecture transcripts for integrated speaking tasks. This aspect will be addressed in more detail in the Detector section.

- 2. Transcription:** To compare the similarity of the contents, the audio recordings of the responses are first transcribed. In principle, any ASR can be used for this step. For

Figure 1
AutoSSD Pipeline



reasons outlined in the next section, we used AWS Transcribe for AutoSSD.

3. Detector: The detector is the main component of AutoSSD. It calculates all possible pairwise response-response and response-prompt similarities and flags pairs whose similarity exceeds a certain threshold. It will be described in detail in the Detector section.

4. Flagged pairs: The detector output is a so-called similarity file containing the flagged response-response and response-prompt pairs. In addition, each pair is assigned a cluster ID. A cluster consists of all pairs that have at least one response in common. The identified clusters can be inspected by the human reviewers, providing insights into how and where specific templates circulate.

5. Human-in-the-Loop Review: Deciding what kind and degree of similarity constitutes cheating depends on the specific items, test program, and the corresponding policy. As such, AutoSSD has a built-in human-in-the-loop component to ensure each pair of transcripts flagged by the detector is reviewed by a human expert. This process is assisted through an interactive web app/dashboard that displays the two suspicious responses next to each other, highlights matching segments, and provides summary statistics, such as the lengths of both responses and different similarity measures. Since the exact interface and functionality of the dashboard can differ significantly depending on the needs and preferences of its users, we will not go into further details in this paper.

6. Decision: Based on the expert reviews, a decision is made on whether test scores

should be released, held, adjusted, canceled, etc. This decision is informed by AutoSSD but is ultimately made by human experts strictly following the test program's policy.

In the following, we will focus on the transcription and detector blocks, which are at the core of the AutoSSD system. While the dashboard is also an integral component, as it facilitates human reviews of the flagged cases, its exact interface and functionality can vary significantly depending on the item types, plagiarism guidelines and personal preferences of the reviewers.

Transcription

As part of the general progress of machine intelligence, Automatic Speech Recognition (ASR) has made tremendous advances in recent years (Prabhavalkar et al., 2023; D. Wang et al., 2019) and, in certain scenarios, matches or even exceeds the accuracy of human experts (Radford et al., 2022). These developments enable us to detect speech similarity based on automatically generated transcripts.

We identified three key requirements for ASR in AutoSSD:

Accuracy: If the ASR is not accurate enough, it can reduce the detection rate of the subsequent detection step. For example, differences in accents or pronunciations can lead to the ASR system making different transcription mistakes for different test takers, thus reducing the similarity between the transcripts. As virtually all takers of language tests are non-native speakers and, in the case of at-home administration, record their responses using non-standardized equipment, this *transcription noise* cannot be eliminated. However, it should be kept as small as possible.

Efficiency: A single administration of a large test can include 10 000 or more test takers. Assuming that the average response length is 45 s, a conservative estimate, approximately 125 hours of speech need to be transcribed per item. In order to ensure the scoring is completed on time, these transcripts must be available within hours, placing a high demand on the efficiency of the ASR system.

Cost: In the end, the accuracy and efficiency of an ASR system need to be

evaluated in light of its economic feasibility. This includes short-term costs, such as hardware or cloud resources, and long-term costs, such as the labor and expertise required to develop, run, maintain, and update the system.

Based on these requirements, we investigated five ASR options, namely:

ETS SpeechRater is developed by ETS and “uses advanced speech recognition and analysis technology to provide detailed feedback about a nonnative speaker’s English-speaking proficiency.” (ETS, 2025a) Although its primary purpose is scoring, it also provides transcripts of the spoken responses. (L. Chen et al., 2018)

ETS Rater Feature Services (RFS) is a service based framework that provides access to various raters and engines, including a transcription service that is an updated and modified variant of the one included in ETS SpeechRater.

Otter is a transcription and note-taking tool developed by Otter.ai. It is geared towards transcribing conversations and generating captions in real time but also offers high-quality general ASR services.

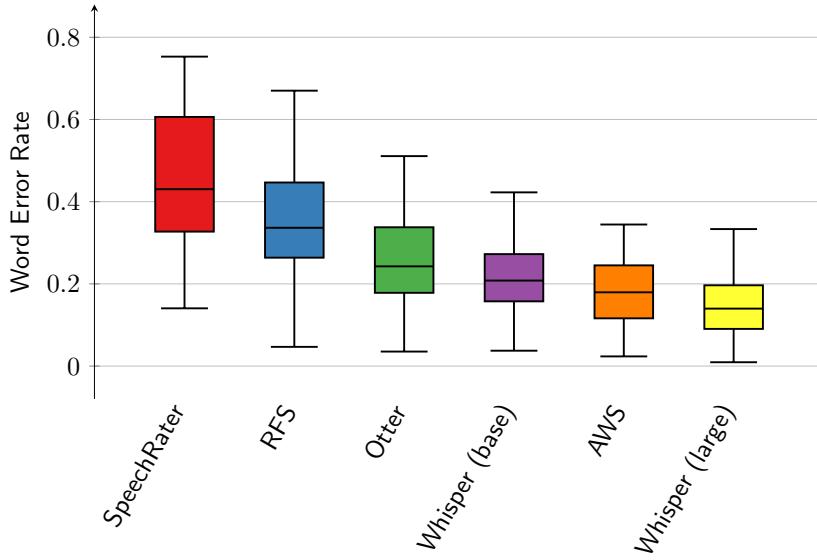
AWS Transcribe is a cloud-based, general-purpose ASR service provided by Amazon Web Services (AWS) (Amazon Web Services, 2017).

Whisper is an ASR system developed by OpenAI that uses large, pre-trained transformer networks that have recently been made publicly available. OpenAI claims that Whisper “approaches human level robustness and accuracy on English speech recognition.” (OpenAI, 2022)

In order to establish the transcription accuracy in the context of AutoSSD, we used audio recordings of 30 matching response pairs identified by human experts. This data set contained 43 unique responses since several occurred in multiple pairs. A human expert transcribed each response, and the resulting transcripts were used as the ground truth¹ to

¹ As the quality of ASR systems improves, this assumption can become problematic. We observed that, in some cases, the automatic transcriptions are *more accurate* than the human ones. In particular, words or segments that were marked as “unintelligible” by the human transcribers were sometimes correctly identified by the machines. Nevertheless, especially in the context of language tests, we believe that “being close to what humans understand” is a viable criterion for the accuracy and usefulness of an ASR system.

Figure 2
Word-Error-Rate Box Plots of ASR Systems



Note. Results are based on 30 similar response pairs with 43 unique responses.

evaluate the ASR systems. Although this data set is small, it provided some useful insights:

- Whisper and AWS Transcribe showed the highest accuracy in our tests, with average word error rates (WERs) between 10 % and 20 %. Note that, although these numbers might seem high, for most practical purposes, the transcripts are virtually identical since the differences are mainly due to variations of the same word, such as plural forms (“house” vs. “houses”) or short forms (“because” vs. “cause”).
- It is important to emphasize that in this experiment the quality of the ASR systems is evaluated in light of the intended use-case of similarity detection. For example, we found that one of the reasons why SpeechRater and RFS showed a high WER in our test is that their transcriptions are in fact closer to the spoken response. That is, they include fillers, such as *hm*, *ehm*, *uh*, and word fragments, such as *the pro... professor argues that...*. In contrast, systems developed solely for the purpose of speech recognition tend to “clean” the transcript. For similarity detection, this cleaning is useful since fillers and word fragments are not part of the underlying templates that we seek to identify. However, for speech *rating*, this information can be highly useful.

- State-of-the-art ASR systems require substantial computing power. In particular, the Whisper networks require modern GPUs and large-scale parallelization to meet the efficiency requirements of AutoSSD. Since AWS Transcribe is a cloud-based service, it does not run into such hardware limitations – at the time of writing, it can process up to 100 jobs in parallel (Amazon Web Services, 2019) – but the service’s running costs must be considered.

Based on these findings, we decided to use AWS Transcript in AutoSSD. It showed the second-highest accuracy in our test, can run efficiently and parallelized in the AWS cloud, integrates well with other AWS-based ETS services, and is available at a competitive price. Finally, the AWS environment used by ETS already adheres to ETS’s strict privacy requirements, which is critical when dealing with test takers’ responses.

Detector

Once the test takers’ responses have been transcribed, the transcripts and the prompt texts of the respective items are handed over to the AutoSSD detector; compare Figure 1. The detector is the core component of AutoSSD. It calculates various similarities between responses and prompts, filters out pairs with suspiciously high similarity, and finally creates a similarity file with all flagged pairs and some additional metadata. In what follows, the tokenizers, similarity measures, and thresholds used for different types of pairs are summarized, and some implementation details are discussed.

Tokenizer

The first step of the detection process is to tokenize the transcripts and prompts. The AutoSSD tokenizer is based on the default scikit-learn (Pedregosa et al., 2011) tokenizer after applying the following text-cleaning steps:

- Remove punctuation marks, including dashes, parentheses, brackets, etc.
- Convert all letters to lowercase.
- Convert text to a list of words/tokens.
- Remove all stop words from this list.

To clarify the last bullet point, stop words are words that provide little to no similarity information. Common examples are frequently used words, such as *a, the, this, that, is, are, etc.*. In the context of language assessment, stop words can also be item-dependent. For example, it can be helpful to ignore words or entire phrases used in the prompt since test takers tend to repeat the prompt or borrow its vocabulary.

Similarity Measures

AutoSSD supports a variety of similarity measures, that is, different ways of defining the similarity of two token sequences. The three similarities currently used in operation are:

- Trigram cosine similarity
- Fuzzy similarity/Token Set Ratio
- Match ratio

These similarities and the advantages and disadvantages of using them for speech similarity detection are summarized below.

Trigram Cosine Similarity

AutoSSD uses the trigram cosine similarity (Manning et al., 2008, Section 6.3) with term frequency-inverse document frequency (TF-IDF) weighting (Robertson, 2004; Spärck Jones, 1972). Different variations of cosine similarity are frequently used in text processing and plagiarism detection (Gomaa & Fahmy, 2013). The pros and cons of using it for speech similarity detection are listed below.

Pros:

- + It can be calculated efficiently. Since all required operations are easily vectorized, and the respective vectors are typically sparse, it is possible to calculate the trigram cosine similarity of millions of response pairs in seconds.
- + The trigram cosine similarity is known to be relatively robust and well-aligned with how human experts rate text similarity. That is, the higher the cosine similarity, the more similar the two documents appear to a human. This “monotonicity” is useful since it justifies the practice of only processing/reviewing pairs whose trigram cosine similarity

exceeds a certain threshold.

Cons:

— The trigram cosine similarity is sensitive against minor spelling and wording differences. While the AI-generated transcripts are free of typos, such differences might still occur. For example, non-native speakers might omit the plural -s in nouns or the third-person -s in verbs. As mentioned above, ASR systems tend to correct such errors, but do not do so consistently. Another source of transcription noise are homophones, such as “sea” and “see,” which are treated as two entirely different words/tokens by the trigram cosine similarity. This issue is one of the motivations for additionally using a fuzzy similarity, which will be discussed in more detail below.

— Trigrams are not always an ideal choice for capturing text or speech similarity. In some cases, one would rather look for longer shared sequences (4,5,6-grams); in other cases, it makes sense to look at shared tokens, that is, unigrams. Generally, there is no one-fits-all n-gram order.

In practice, the trigram cosine similarity is highly useful. In particular, for the reasons stated above, it is useful for a first filtering, meaning that more complex similarities are only calculated for pairs whose trigram cosine similarity exceeds a certain threshold. This technique reduces the computational load significantly and will be explained in more detail later in this section. Finally, our experiments showed that in most cases, trigrams provide a good trade-off between local and global similarity and that higher or lower n-gram orders added little information.

Fuzzy similarity

The cosine similarity is defined on a token/word level, that is, it captures how many tokens – or sequences of tokens, such as trigrams – two texts have in common. This definition of text similarity is often useful and allows for an efficient calculation. However, it is not robust against some artifacts commonly found in handwritten/hand-typed texts, such as typos or informal spellings of a word. For example, consider the two sentences: (a)

I could not make it, because I fell sick. (b) I could not maje it, cause I fell sick. Here, the second sentence is a “sloppy” version of the first one, where “maje” is a typo – j is next to k on common keyboard layouts – and “because” was shortened to “cause.” AutoSSD tokenizes both sentences to (a) [could, not, make, it, because, fell, sick] (b) [could, not, maje, it, cause, fell, sick] These two token sequences have no trigrams in common, hence, the trigram cosine similarity of the two underlying sentences is zero. The fuzzy similarity aims to detect such segments that do not match on the token level but are highly similar on a character level.

The fuzzy similarity used by AutoSSD combines resources from several libraries. First, the documents are tokenized as described above. In addition, the tokenizer ignores all words in the default list of English stop words of the Natural Language Toolkit (NLTK) for Python (Bird et al., 2009). The similarity of the resulting token sequences is then calculated using the Token-Set-Ratio similarity as implemented by the RapidFuzz library (Bachmann, 2023).

Pros:

- + As mentioned above, the main advantage of fuzzy similarity is that it does not rely on exact token matching. The two sentences above with zero trigram cosine similarity have a fuzzy similarity of 0.78.

- + It is insensitive to reordering sentences or sentence parts. For example, the fuzzy similarity of (a) Oil prices are determined by global supply and demand, rather than any country’s domestic production level. (b) Rather than any country’s domestic production level, global supply and demand determine oil prices. is larger than 0.99, while their trigram cosine similarity is merely 0.54.

- + It can catch similarities even if two texts are of vastly different lengths. For example, assume that the first text is a copy of the first paragraph of a second, much longer text. Since all tokens of the first text are contained in the second essay, the fuzzy similarity of both is one. In contrast, the cosine similarity will be small since the absolute

number of shared tokens between both essays is small.

Cons:

- Not taking the order in which words/tokens appear into account is a strength and a weakness of the fuzzy similarity. In particular, it can lead to high similarities between responses that use a similar vocabulary but are semantically different. For example, the sentences (a) `Rover likes to play ball in the dog park.` (b) `I'd like to have a ball park figure on the Range Rover.` have a fuzzy similarity of 0.76, while their trigram cosine similarity is zero.
- Similarly, the fact that a small overlap of words/tokens is sufficient for a high fuzzy similarity can cause problems. For example, if a test taker only repeats the prompt without answering it, their response is highly similar to that of all test takers who also repeat the prompt but then go on to provide a proper answer.
- Finally, the fuzzy similarity is difficult to interpret and may not align well with expert judgment. In particular, it often overestimates the similarity of response pairs, making it hard for human reviewers to pinpoint which aspects or segments of the responses led to this high value.

In practice, it turned out that the fuzzy similarity is helpful for identifying response-prompt pairs but much less so for identifying response-response pairs; see the Results and Findings section for more details. This is in line with the properties discussed above, namely, that the response of a test taker repeating the prompt will likely have a high response-prompt fuzzy similarity. However, the question why fuzzy similarity is not a good choice for flagging response-response pairs needs further investigation. One reason could be that, since AutoSSD works on machine-generated transcripts instead of human-written essays, the compared texts rarely contain typos and informal spellings.

Match Ratio

When analyzing the cases detected by ETS's *essay* similarity detector, AutoESD (Choi et al., 2024), it became clear that the match ratio showed the best separation

between true and false positives. Therefore, we decided to include the match ratio in the AutoSSD detector as a third similarity measure.

The match ratio similarity uses the SequenceMatcher class of Python’s difflib library (Foundation, 2023). The latter can identify matching segments in two sequences. To this end, again, both documents are first tokenized. The longest contiguous matching subsequence of tokens is identified and added to a list. The algorithm then identifies the longest matching subsequence of tokens before and after the already identified subsequence and adds them to the list as well. It proceeds in this recursive manner until no more matching subsequences can be found. The match ratio is then given by the ratio of twice the total length of all matching segments to the total lengths of both sequences. See the difflib documentation for more details (Foundation, 2023).

Pros:

- + As mentioned above, the match ratio aligns well with the judgment of human experts – even better than the trigram cosine similarity. A response pair with a higher match ratio is almost always “more similar” than a pair with a lower match ratio.
- + The match ratio is an approximate measure for the overlap of two texts. The latter is often used as a criterion to establish plagiarism. For example, many scientific journals limit the permissible overlap with the literature to 20 % to 30 % (IEEE, 2023).
- + The match ratio can easily be visualized by highlighting all matching segments in the underlying texts. This makes it a useful tool for the human reviewers when comparing a response to a potential source.

Cons:

- Owing to the underlying recursive search, the match ratio misses matches that appear before the longest matching segment in one sequence and after the longest matching segment in the other sequence. For example, the two sentences given in the second “pro” bullet point of the fuzzy similarity have a match ratio of only 0.47, because the longest matching segment appears at the end of the first sentence, but at the beginning

of the second sentence.

- The match ratio is one of the more expensive similarity measures to calculate, and its computation time grows quadratically with the essay length (Foundation, 2023).
- Finally, the fact that the match ratio is easy to visualize is also a potential danger. Namely, reviewers might give it disproportionate weight in their decision whether a certain degree of similarity constitutes cheating. In fact, since its introduction, the AutoSSD dashboard has had the feature to highlight the matching segments underlying the definition of the match ratio. Therefore, there is a non-negligible possibility that the match ratio was identified as the most salient feature simply because reviewers – consciously or unconsciously – tend to base their decision on it. This aspect will be further investigated in the future.

The match ratio has proven helpful, particularly in identifying response-response pairs. For the latter, it produces fewer false positives than other similarities and provides useful, interpretable information to the reviewers. It is less helpful in identifying response-prompt pairs, likely because, in contrast to the fuzzy similarity, it is affected by one document being significantly shorter than the other.

Parameters and Thresholds

So far, we have detailed the tokenizer and the three similarities AutoSSD uses. This subsection will discuss how these similarities are used to flag response-response and response-prompt pairs.

All response-response and response-prompt pairs are first filtered by trigram cosine similarity. As explained above, the latter provides a good baseline and can be calculated efficiently. Typically, the vast majority of pairs are already discarded based on their cosine similarity. The remaining pairs are further filtered by fuzzy similarity and match ratio. Only pairs exceeding all three similarity thresholds are flagged for human review.

The particular thresholds that the similarities of response pairs need to exceed in order for them to get flagged depend on the item type as well as the pair type

(response-response or response-prompt). They were established based on an analysis of reviewed pairs and feedback from expert reviewers. However, since cheating techniques evolve continuously and test takers adapt to the introduction of systems like AutoSSD, these thresholds require periodical adjustments.

Results and Findings

Since the operationalization of AutoSSD, we have logged the outcomes of many expert reviews, which allow us to gain some insight into the performance of AutoSSD and the usefulness of the similarity measures discussed in the previous section.

The results presented here are based on TOEFL iBT tests (ETS, 2025b) administered between November 2022 and March 2023. The data are split into true and false positives. True positives are pairs flagged by AutoSSD whose subsequent expert review resulted in a complete or partial score cancellation. False positives are pairs flagged by AutoSSD whose review did not result in a score cancellation. Unfortunately, we cannot make statements about the true and negative rates since only flagged pairs underwent human review. As a consequence, our performance evaluation is limited to the *precision* of the detector, that is, the share of true positives among all flagged pairs.

Note that the categories “true positive” and “false positive” should be taken with caution since they do not adequately reflect the many gray areas in between. For example, even response pairs that are ultimately released may still contain suspicious segments that warrant closer inspection by a human expert.

The true and false positives in our data set are summarized in Table 1 for different item and pair types. Here, independent items consist of short standalone prompts that ask the test taker to express their experiences, ideas, and opinions on a certain topic. Integrated items combine speaking with listening and/or reading tasks, thus testing a broader spectrum of language skills. Typically, an integrated item asks test takers to compare and contrast the content of a reading passage to that of a short lecture or

Table 1
AutoSSD Precision

Item and Pair Type	Precision
Independent Response-Response	29 %
Integrated Response-Response	48 %
Integrated Response-Prompt	64 %

dialogue. Interested readers can find more details on the official TOEFL website.² The low precision for independent items is largely due to the first months of operation when common prompt language in the responses caused many false positives. These numbers decreased significantly when we started filtering prompt language in the similarity calculations. Finally, note that since independent speaking items have a short prompt that test takers are free to repeat in the response, no response-prompt similarity is calculated for this item type.

In a general machine learning context, the performance metrics in Table 1 may seem alarmingly low. However, in the context of high-stakes testing, there are some key differences:

- We deliberately chose the detection thresholds to minimize the false negative rate, that is, the number of undetected cheating attempts. This strategy is critical to ensure test integrity but necessarily comes at the cost of an increased false positive rate.
- Again, we would like to underscore that all flagged pairs undergo a review by human experts. In fact, it can be argued that the main objective of AutoSSD is to reduce the number of pairs that undergo an expert review to a manageable level.
- To put the numbers in Table 1 into perspective, note that a precision of 50 % means that every second expert review results in a score cancellation. In the context of high-stakes testing, where every suspicious event or behavior needs to be investigated, such a rate is considered acceptable or even high.

² <https://www.ets.org/toefl/test-takers/ibt/about/content/speaking.html>

- We continuously adjust thresholds and parameters like stop words based on feedback and data from expert reviews. The presented results are averaged over several months, with the precision being lower during the early periods of running AutoSSD operationally.

- Finally, we explicitly do not target a perfect precision but values of around 50 %. In doing so, we create a growing, balanced, high-quality data set of flagged, reviewed, and labeled response pairs. Such a data set is highly valuable for many research purposes in test security and other areas.

In summary, AutoSSD is an instructive example of a human-in-the-loop AI system that combines the strengths of humans and machines and improves its performance iteratively. While humans are still unmatched in their ability to understand and holistically evaluate test takers' responses, it requires AI technology to pre-screen and filter millions of potentially suspicious response pairs. In turn, the labels assigned by the expert reviewers provide valuable data for adjusting similarity thresholds and assessing the usefulness of different similarity metrics, thus closing the human-AI loop.

Conclusions and Outlook

AutoSSD, as a human-in-the-loop AI system, has proven to be highly impactful in operational use. In particular, the similarity-measure-based approach balances three important goals: catching a wide variety of cheating attempts, being computationally feasible, and providing helpful information to human experts. Of course, alternative approaches exist, but, from our experience, often fall short in at least one of the three criteria. For example, more sophisticated plagiarism detection tools that operate on multiple levels (Arabi & Akbari, 2022) and perform synonym substitution or text alignment (Sanchez-Perez et al., 2015) are likely to catch more elaborate cheating attempts that the current AutoSSD system misses. However, such tools typically run orders of magnitude slower, have many more parameters requiring tuning, and are more sensitive to item and response type changes. On the other hand, end-to-end solutions based on trained

or fine-tuned neural networks provide little to no information to human reviewers and require large sets of high-quality training data. In our use case, this approach often faces a chicken-and-egg dilemma, as acquiring ample training data is nearly impossible without an existing automated speech similarity detector.

The modular design of AutoSSD also makes it easy to adapt the system to different tests and items, for example, by modifying the thresholds or by adding or replacing similarity metrics. Both can typically be done based on small sets of training data in combination with expert knowledge and feedback.

Nevertheless, the current version of AutoSSD is not free from issues requiring further improvements and extensions that are being actively researched at ETS. For example, while using ASR systems to reduce the problem of quantifying speech similarity to that of quantifying text similarity is an elegant and efficient approach, it also comes with limitations. Human experts often base their decisions not only on the transcripts but also on the original audio recordings. The latter contain much more information and can reveal, for example, inconsistencies between language use and speaking proficiency or speaking patterns that are indicative of reading from a prepared template. An AI-powered system that can extract useful features directly from the audio recordings and fuse them with the AutoSSD output is currently being investigated. Another ongoing strand of research is to supplement the similarity-based detector with an AI classifier that considers more features. Such a system can be used to inspect the flagged cases and “unflag” likely false positives, thus reducing the workload of the human reviewers. Again, note that such a system cannot replace the detector since it relies on the latter to produce training data and to reduce the number of cases to a feasible level.

References

- Amazon Web Services. (2017). Amazon Transcribe – Speech to Text. Retrieved May 15, 2025, from <https://aws.amazon.com/transcribe/>
- Amazon Web Services. (2019). *Amazon Transcribe now Supports Job Queuing for Batch Workloads*. Retrieved March 11, 2025, from <https://aws.amazon.com/about-aws/whats-new/2019/12/amazon-transcribe-supports-job-queuing-batch-workloads/>
- Arabi, H., & Akbari, M. (2022). Improving plagiarism detection in text document using hybrid weighted similarity. *Expert Systems with Applications*, 207, 118034. <https://doi.org/https://doi.org/10.1016/j.eswa.2022.118034>
- Bachmann, M. (2023). *RapidFuzz*. Retrieved June 30, 2023, from <https://github.com/maxbachmann/RapidFuzz>
- Bilen, E., & Matros, A. (2021). Online cheating amid COVID-19. *Journal of Economic Behavior & Organization*, 182, 196–211. <https://doi.org/10.1016/j.jebo.2020.12.004>
- Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: Analyzing text with the natural language toolkit*. O'Reilly.
- Chen, L., Zechner, K., Yoon, S.-Y., Evanini, K., Wang, X., Loukina, A., Tao, J., Davis, L., Lee, C. M., Ma, M., Mundkowsky, R., Lu, C., Leong, C. W., & Gyawali, B. (2018). Automated scoring of nonnative speech using the SpeechRater v. 5.0 engine. *ETS Research Report Series*, 2018(1), 1–31. <https://doi.org/10.1002/ets2.12198>
- Chen, Y., Lee, Y.-H., & Li, X. (2022). Item pool quality control in educational testing: Change point model, compound risk, and sequential detection. *Journal of Educational and Behavioral Statistics*, 47(3), 322–352. <https://doi.org/10.3102/10769986211059085>
- Choi, I., Hao, J., Li, C., Fauss, M., & Novák, J. (2024). AutoESD: An automated system for detecting nonauthentic texts for high-stakes writing tests. *ETS Research Report Series*. <https://doi.org/https://doi.org/10.1002/ets2.12383>

- ETS. (2025a). *The SpeechRater Service*. Retrieved March 7, 2025, from
<https://www.ets.org/speechrater.html>
- ETS. (2025b). *TOEFL iBT Test*. Retrieved March 11, 2025, from
<https://www.ets.org/toefl/institutions/ibt.html>
- Evanini, K., & Wang, X. (2014). Automatic detection of plagiarized spoken responses.
Proceedings of the Ninth Workshop on Innovative Use of NLP for Building Educational Applications, 22–27. <https://doi.org/10.3115/v1/w14-1803>
- Foltýnek, T., Meuschke, N., & Gipp, B. (2019). Academic plagiarism detection: A systematic literature review. *ACM Comput. Surv.*, 52(6).
<https://doi.org/10.1145/3345317>
- Foundation, P. S. (2023). *Difflib*. Retrieved July 5, 2023, from
<https://docs.python.org/3/library/difflib.html>
- Gomaa, W. H., & Fahmy, A. A. (2013). A survey of text similarity approaches.
International Journal of Computer Applications, 68(13), 13–18.
<https://research.ijcaonline.org/volume68/number13/pxc3887118.pdf>
- IEEE. (2023). *Introduction to the Guidelines for Handling Plagiarism Complaints*.
Retrieved July 10, 2023, from
<https://www.ieee.org/publications/rights/plagiarism/plagiarism.html>
- Janke, S., Rudert, S. C., Petersen, Å., Fritz, T. M., & Daumiller, M. (2021). Cheating in the wake of COVID-19: How dangerous is ad-hoc online testing for academic integrity? *Computers and Education Open*, 2, 100055.
<https://doi.org/10.1016/j.caeo.2021.100055>
- Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.
- Newton, P. M., & Essex, K. (2023). How common is cheating in online exams and did it increase during the COVID-19 pandemic? A systematic review. *Journal of Academic Ethics*, 1–21. <https://doi.org/10.1007/s10805-023-09485-5>

Noorbehbahani, F., Mohammadi, A., & Aminazadeh, M. (2022). A systematic review of research on cheating in online exams from 2010 to 2021. *Education and Information Technologies*, 27, 8413–8460. <https://doi.org/10.1007/s10639-022-10927-7>

OpenAI. (2022). *Introducing Whisper*. Retrieved March 7, 2025, from <https://openai.com/index/whisper/>

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830. <https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html>

Prabhavalkar, R., Hori, T., Sainath, T. N., Schlüter, R., & Watanabe, S. (2023). End-to-end speech recognition: A survey.

<https://doi.org/10.1109/taslp.2023.3328283>

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2022). Robust speech recognition via large-scale weak supervision.

<https://doi.org/10.48550/arXiv.2212.04356>

Robertson, S. (2004). Understanding inverse document frequency: On theoretical arguments for IDF. *Journal of Documentation*, 60(5), 503–520. <https://doi.org/10.1108/00220410410560582>

Sanchez-Perez, M. A., Gelbukh, A., & Sidorov, G. (2015). Dynamically adjustable approach through obfuscation type recognition. *CEUR Workshop Proceedings*, 1391. <https://ceur-ws.org/Vol-1391/92-CR.pdf>

Spärck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1), 11–21. <https://doi.org/10.1108/eb026526>

St-Onge, C., Ouellet, K., Lakhal, S., Dubé, T., & Marceau, M. (2022). COVID-19 as the tipping point for integrating e-assessment in higher education practices. *British*

- Journal of Educational Technology*, 53(2), 349–366.
<https://doi.org/10.1111/bjet.13169>
- Wang, D., Wang, X., & Lv, S. (2019). An overview of end-to-end automatic speech recognition. *Symmetry*, 11(8). <https://doi.org/10.3390/sym11081018>
- Wang, X., Evanini, K., Bruno, J., & Mulholland, M. (2016). Automatic plagiarism detection for spoken responses in an assessment of English language proficiency. *2016 IEEE Spoken Language Technology Workshop (SLT)*, 121–128.
<https://doi.org/10.1109/SLT.2016.7846254>
- Wang, X., Evanini, K., Mulholland, M., Qian, Y., & Bruno, J. V. (2019). Application of an automatic plagiarism detection system in a large-scale assessment of English speaking proficiency. In H. Yannakoudakis, E. Kochmar, C. Leacock, N. Madnani, I. Pilán, & T. Zesch (Eds.), *Proceedings of the fourteenth workshop on innovative use of nlp for building educational applications* (pp. 435–443). Association for Computational Linguistics. <https://doi.org/10.18653/v1/W19-4445>
- Wang, X., Evanini, K., Qian, Y., & Zechner, K. (2019). Using very deep convolutional neural networks to automatically detect plagiarized spoken responses. *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 764–771.
<https://doi.org/10.1109/ASRU46091.2019.9003924>
- Yoon, S.-Y., & Xie, S. (2014). Similarity-based non-scorable response detection for automated speech scoring. *Proceedings of the Ninth Workshop on Innovative Use of NLP for Building Educational Applications*, 116–123.
<https://doi.org/10.3115/v1/w14-1814>
- Zheng, M., Bender, D., & Lyon, C. (2021). Online learning during COVID-19 produced equivalent or better student course performance as compared with pre-pandemic: Empirical evidence from a school-wide comparative study. *BMC medical education*, 21, 1–11. <https://doi.org/10.1186/s12909-021-02909-z>