Customizing `rules.tsv`: Teaching AI What to Detect

`rules.tsv` is AI's phrase-classification table.

It does not train Moonshine and it does not modify the acoustic model. It works after ASR:

```
Moonshine transcript
    ↓
Normalize (lowercase, remove punctuation and apostrophes)
    ↓
Match against phrases in rules.tsv
    ↓
Accumulate weights per label
    ↓
Output H / VM / S / T / IVR / Unknown
    ↓
Policy decides: play human message, leave voicemail,
play screening reply, wait for transfer, or hang up
```

In other words, customizing `rules.tsv` means:

> Tell the program which words should make it treat the caller as a human, voicemail, screening assistant, transfer prompt, or IVR.

---

1. The Five Labels

Label	Meaning	Typical action
`H`	Live human	Play the human message
`VM`	Voicemail / answering machine	Enter the voicemail flow
`S`	Screening assistant / virtual receptionist	Play the screening reply
`T`	Transfer / hold prompt	Say nothing and wait for the human
`IVR`	Keypad-based auto attendant	Hang up or enter the IVR flow

The numeric values in the log are:

```
0 = Unknown
1 = H
2 = VM
3 = S
4 = IVR
5 = T
```

For example:

```
label=3  → Screening
label=1  → Human
label=5  → Transfer
```

---

2. File Format

Each line is one rule. Columns must be separated by **Tab** characters, not spaces:

```
label<Tab>weight<Tab>phrase
```

Visually:

```
S       4       who is calling
H       4       this is he
VM      5       leave a message after the tone
T       5       please hold while i connect you
IVR     5       press 1
```

Rules:

- The first column must be `H`, `VM`, `S`, `T`, or `IVR`.
- The second column is a numeric weight. Integers and decimals are allowed.
- The third column is the phrase.
- Lines beginning with `#` are comments.
- Blank lines are ignored.
- A rule line must contain exactly two Tab separators. Otherwise it is skipped.
- An inline `#` is not a comment. It becomes part of the phrase.
- The file should be saved as UTF-8.

Wrong examples:

```
S 4 who is calling          ← spaces instead of Tabs
S       4       who is calling  # note  ← trailing comment becomes part of the phrase
S       4       who     is      calling   ← extra Tab inside the phrase
```

---

3. Normalization

Matching is performed on normalized text, not on the raw ASR output.

3.1 Lowercase

```
"Who Is Calling?" → "who is calling"
```

3.2 Punctuation Becomes a Space

```
"Hello, who is this?" → "hello who is this"
```

3.3 Apostrophes Are Removed, Not Expanded

```
"who's"  → "whos"
"who is" → "who is"
```

These are therefore different phrases:

```
who's calling
who is calling
```

If both forms are likely, add both:

```
S       4       who's calling
S       4       who is calling
```

Curly apostrophes such as `’` and `‘` are also removed.

3.4 Hyphens Split Words

```
"follow-up" → "follow up"
```

Write the rule as:

```
T       4       follow up
```

3.5 Chinese and Accented Characters Are Preserved

Chinese characters and Spanish accented characters are preserved as UTF-8 bytes.

Examples:

```
S       5       asistente virtual
S       5       这是虚拟助手
```

The rules file and the ASR transcript must use the same UTF-8 character form.

For Spanish, keep accents if the transcript is likely to keep them:

```
VM      5       después del tono
```

Do not write only `despues` unless you have verified that the transcript consistently drops the accent.

---

4. How Matching Works

A rule matches when the normalized phrase occurs in the normalized transcript, padded with spaces:

```
padded.find(" " + phrase + " ") != npos
```

For example, the transcript:

```
you have reached the virtual assistant for mark
```

matches:

```
S       5       virtual assistant
```

It also matches:

```
S       5       assistant
```

But this does not match `virtual assistant`:

```
S       5       virtual assistants
```

because it is a different word.

Longer phrases are usually more reliable

Prefer complete phrases:

```
S       5       you have reached the virtual assistant
S       5       please state your name and the reason for your call
```

Instead of relying only on:

```
S       2       assistant
```

A single broad word can cause false positives. For example, a human may say:

```
"Can I speak to the assistant?"
```

---

5. Weights and Scores

Each label has its own score:

```
score[H]
score[VM]
score[S]
score[T]
score[IVR]
```

Every matched rule adds its weight to that label:

```
S       4       virtual assistant
S       5       please state your name
```

If both match:

```
score[S] = 4 + 5 = 9
```

Suggested Weights

Weight	Use
`1`	Very weak hint; easy to over-trigger
`2`	Common but ambiguous word or phrase
`3`	Fairly specific phrase
`4`	Strong phrase
`5`	Complete sentence that is almost conclusive by itself

In general, do not go above `5` unless there is a clear business reason.

Thresholds

Classification is not "one match wins." The accumulated score must reach a threshold:

```
IVR  score >= 4
T    score >= 4
S    score >= 4 and S >= VM
VM   score >= 4
H    score >= 4, with VM < 3 and S < 3
H    score >= 2, short text (<= 6 words), with VM < 3 and S < 3
otherwise Unknown
```

This means:

- One weight-4 rule is usually enough.
- Two weight-2 rules can also reach 4.
- Conflicting labels can keep the result at Unknown.
- Higher weight is not always better; conflicting rules pull the result in different directions.

There is also an explicit label priority:

```
1. IVR score >= 4              → IVR
2. T   score >= 4              → T
3. S   score >= 4 and S >= VM  → S
4. VM  score >= 4              → VM
5. H   score >= 4 and S < 3 and VM < 3 → H
6. H   score >= 2, short text, S < 3, VM < 3 → H
7. otherwise Unknown
```

For example:

```
S 5 ...
T 4 ...
```

The result is still `T` because transfer is checked before screening. Always check whether a new rule can make another label win earlier.

---

6. Built-in Score Adjustments

The classifier adds several bonuses that are not in `rules.tsv`. Understanding them explains why a rule may behave differently than expected.

6.1 Question adds 1.5 to Screening

If the raw transcript contains `?`:

```
score[S] += 1.5
```

The log shows:

```
S:question +1.5
```

This is why:

```
"Who is calling?"
```

is strongly classified as Screening even with only one `S 4 who is calling` rule.

6.2 Short Greeting adds 1 to Human

If:

- `score[H] > 0`
- the transcript has at most 6 words
- `score[S] < 4`
- `score[VM] < 4`

then:

```
score[H] += 1
```

The log shows:

```
H:short greeting +1
```

This makes short greetings such as `hello` easier to classify as Human.

6.3 IVR Cue adds 2

If any IVR rule matched:

```
score[IVR] += 2
```

because phrases like `press 1` are highly specific.

6.4 Screening/VM Interaction adds 0.5

If:

- `score[S] > 0`
- `score[VM] > 0`
- `score[S] >= score[VM] - 1`

then:

```
score[S] += 0.5
```

6.5 "this is ... assistant" adds 4 to Screening

If the normalized transcript contains both:

```
this is
assistant
```

then:

```
score[S] += 4
```

This catches phrases such as:

```
"Hi, this is Jennifer's assistant."
```

without requiring the word `virtual` or `Google`.

---

7. Example Rules by Label

7.1 Screening Assistant `S`

```
S       5       you have reached the virtual assistant
S       5       virtual assistant for
S       5       this is a virtual assistant
S       5       please state your name and the reason for your call
S       5       state your name and the reason for your call
S       4       name and the reason for your call
S       4       who is calling
S       4       who's calling
S       4       what is this regarding
S       4       how may i assist you
S       4       how can i assist you
S       4       who may i say is calling
```

7.2 Voicemail `VM`

```
VM      5       leave a message after the tone
VM      5       after the beep
VM      4       leave your name and number
VM      4       the person you are calling is not available
VM      4       voice mail
VM      4       voicemail
```

`not available` is common and ambiguous. Do not give it a high weight by itself; prefer a longer phrase.

7.3 Transfer `T`

```
T       5       please hold while i connect you
T       5       i will connect you
T       5       let me connect you
T       5       transferring your call
T       4       one moment
T       4       please hold
T       4       i will let them know
```

`one moment` can also be said by a human. Keep its weight at 4 or lower unless the business requires otherwise.

7.4 Human `H`

```
H       4       this is he
H       4       this is she
H       4       this is mark speaking
H       4       speaking
H       3       how may i help you
H       2       hello
H       2       who is this
```

Do not make `who is this` too strong, because a screening assistant can also ask:

```
"Hello, who is this?"
```

7.5 IVR `IVR`

```
IVR     5       press 1
IVR     5       press one
IVR     5       using your keypad
IVR     4       main menu
IVR     4       for customer service
IVR     3       please listen carefully
```

---

8. Multilingual Examples

English

```
S       5       please state your name and the reason for your call
VM      5       leave a message after the tone
T       5       please hold while i connect you
H       4       this is mark speaking
```

Spanish

```
S       5       asistente virtual
S       5       diga su nombre y el motivo de su llamada
VM      5       deje un mensaje después del tono
T       5       por favor espere mientras le conecto
IVR     5       marque 1
IVR     5       presione 1
H       4       hola
H       4       soy mark
```

Chinese

```
S       5       这是虚拟助手
S       5       请说明您的姓名和来电原因
VM      5       请在提示音后留言
T       5       请稍等，我帮您转接
IVR     5       请按1
IVR     5       请按一
H       4       你好
H       4       我是
```

Chinese text has no spaces. The rules file and transcript are normalized as UTF-8 bytes on both sides. Do not insert spaces between Chinese characters unless the ASR output actually contains them.

---

9. Editing and Applying Rules

Step 1: Back up

```
Copy-Item rules.tsv rules.tsv.bak
```

Step 2: Edit

Use an editor that supports UTF-8 and shows whitespace, such as Notepad++:

- select UTF-8 encoding;
- enable "Show Whitespace and Tab";
- confirm columns are separated by Tab, not spaces;
- do not leave trailing content on rule lines.

Step 3: Put the file in the runtime directory

The rules file can be placed at:

```
<program directory>\rules.tsv
```

Then verify that `Rules file` in AI Settings points to it.

It can also be configured in the Profile:

```
<sirules>...</sirules>
```

The log prints the file actually loaded:

```
rules=C:\...\rules.tsv
```

Always check this path. Editing a different copy has no effect.

Step 4: Restart

`rules.tsv` is loaded only during initialization:

```
si_global_init_moonshine(..., rules_path, ...)
```

There is no hot reload. Completely exit the program and start it again.

Step 5: Read the log

A verdict looks like:

```
label=3 conf=0.99 text="..."
cues=S:virtual assistant +5, S:question +1.5
```

Check:

- Does `label` match the expected label?
- Does `cues` contain the rules you added?
- Is the score high enough to cross the threshold?
- Is another label winning earlier?

---

10. Troubleshooting

Symptom	Likely cause	Fix
Rule never matches	Spaces instead of Tab	Use Tab separators
Rule never matches	Punctuation/apostrophe difference	Compare normalized forms
Rule matches but label is wrong	Another label has a higher score	Read `cues`, adjust weights
Label is Unknown	Score does not reach threshold	Add a full phrase or increase weight
H triggers too easily	`hello` / `who is this` is too strong	Lower weights; add more specific H phrases
S is misclassified as H	Short-greeting bonus and H rules are too strong	Add complete S phrases; avoid relying on `assistant` alone
VM is misclassified as S	Both S and VM match	Inspect the S/VM priority and add complete VM phrases
Changes have no effect	Program is still using another rules file	Check `rules=` in the log
Program uses old rules	Not restarted	Fully exit and restart

---

11. What `rules.tsv` Cannot Fix

`rules.tsv` only answers "what does this text look like?" It does not control dialog timing or playback.

11.1 `NO_HUMAN` while the call is still ringing

This is an application startup-timing issue or a policy probe-window issue, not a rules issue. It must start only after the call reaches `CONFIRMED`.

11.2 `PLAYBACK_TIMEOUT`

This is a playback-completion callback problem, or a shared playback state problem. Rules cannot fix it.

11.3 `SCREENED_NO_TRANSFER`

This is the policy waiting for a human transfer and timing out. It is not a classification error.

11.4 Human detected but no action was executed

This is a policy state-machine or callback problem, not a rules problem.

11.5 The transcript itself is wrong

If Moonshine mishears the words, better rules will not help. Check:

- whether the model matches the language;
- 8 kHz to 16 kHz resampling;
- recording quality;
- whether the phrase is actually present.

Do not use rules to hide an audio or model problem.

---

12. A Safe Tuning Workflow

1. Back up the current `rules.tsv`.
2. Add only one to three highly specific phrases.
3. Restart and collect 10–20 real calls.
4. Inspect `label` and `cues` in the log.
5. Adjust only the rules involved in actual mistakes.
6. Do not add a large number of single-word rules at once.
7. Do not raise `assistant`, `hello`, or `who is this` to weight 5 just to fix one case.
8. Test concurrency, ringing, and speech-during-playback separately.

---

13. Minimal Template

```
# label<Tab>weight<Tab>phrase

# Screening assistant
S       5       you have reached the virtual assistant
S       5       please state your name and the reason for your call
S       4       who is calling

# Voicemail
VM      5       leave a message after the tone
VM      4       the person you are calling is not available

# Transfer
T       5       please hold while i connect you
T       4       one moment

# Human
H       4       this is he
H       4       this is she
H       4       speaking
H       2       hello

# IVR
IVR     5       press 1
IVR     5       press one
IVR     4       main menu
```

Summary

The goal is not to add as many rules as possible. The goal is to:

```
express strong signals with complete phrases
express weak signals with low weights
avoid conflicts between labels
verify every change with real log cues and scores
```

Rules decide classification only.

Playback, waiting, transfer, hangup, and retries are still controlled by the policy and the application callbacks.
