MuseScore mscz-concatenator:

How Fuzzy part matching works:

MATCH Strategy: (strict)
Requires exact number matching
"Violin 1" vs "Violin 2" = 0.0 (immediate failure)
"1st Violin" vs "Violin I" = 0.0 (different number representations)
"Flute" vs "Flute 1" = 0.0 (one has number, one doesn't)

IGNORE Strategy: (liberal)
Completely ignores numbers
"Violin 1" vs "Violin 2" = 1.0 (both are "Violin")
"1st Violin" vs "Violin 3" = 1.0 (numbers ignored)
"Flute" vs "Flute 1" = 1.0 (numbers ignored)

PREFER Strategy (balanced):
"Violin 1" vs "Violin 1" = 1.0 (perfect match)
"Violin 1" vs "Violin 2" = 0.75 (penalty applied)
"1st Violin" vs "Violin I" = 0.75 (penalty applied)

So MATCH is for when part numbers must be exact, while IGNORE treats all variations of an instrument as the same regardless of numbering.

Technical:

fuzzy.py is part of Leon Dionne’s mscore python library.

1. Exact Match = 1.0

```
if self.ref == name:
    return 1.0
```

2. Word-by-Word Comparison

Split both names into words
Compare each word from the longer list with each word from the shorter list
Take the best match for each word

3. Word Scoring Rules (from _word_score()):

Exact word match: 1.0 ("violin" vs "violin")
Plural variations: 0.75 ("violin" vs "violins", "cello" vs "cellos")
No match: 0.0 ("violin" vs "flute")

4. Number Handling (based on strategy):

PREFER: If numbers don't match, multiply score by 0.75
MATCH: If numbers don't match, return 0.0 immediately
IGNORE: Ignore number differences completely

5. Final Score Calculation:

```
# Average of all word scores, with number penalties applied
f_score = sum(scores) / len(scores)
if numbers_strategy == PREFER and num1 != num2:
    return f_score * 0.75  # Apply number penalty
```

Examples with Different Thresholds:
Example 1: "Violin 1" vs "1st Violin"

Words: ["violin", "1"] vs ["1st", "violin"]
"violin" vs "violin" = 1.0
"1" vs "1st" = 0.75 (number match via fuzzy number logic)
Average = ~0.875
Threshold 0.7: PASS
Threshold 0.9: FAIL

Example 2: "Violins" vs "Violin Section"

Words: ["violins"] vs ["violin", "section"]
"violins" vs "violin" = 0.75 (plural match)
"violins" vs "section" = 0.0
Best scores: [0.75, 0.0] → Average = 0.375
Any threshold > 0.375: FAIL

Example 3: "Cello" vs "Violoncello"

Words: ["cello"] vs ["violoncello"]
No match = 0.0
Any threshold > 0.0: FAIL

Practical Threshold Guidelines:

0.9: Very strict (almost exact matches only)
0.7: Moderate (allows common variations)
0.5: Liberal (allows significant differences)
0.3: Very liberal (matches most instrument families)

The threshold is applied to each part individually - ALL parts must meet or exceed the threshold for the file to be considered compatible.

This gives users fine control over how strict they want the matching to be!
