Chapter 2

Basic Text Processing

Regular Expressions
Regular expressions

• A formal language for specifying text strings
• How can we *search* for any of these?
  • woodchuck
  • woodchucks
  • Woodchuck
  • Woodchucks
  • Ill vs. illness
  • color vs. colour

Example

• Does $>$ *grep* “elect” *news.txt* return every line in a file called *news.txt* that contains the word “elect”

  ```
  elect
  ```

  Misses capitalized examples

  ```
  [eE]lect
  ```

  Incorrectly returns select or electives

  ```
  [^a-zA-Z][eE]lect[^a-zA-Z]
  ```
Errors

• The process we just went through was based on fixing two kinds of errors
  • Matching strings that we should not have matched (there, then, other)
    • False positives (Type I)
  • Not matching things that we should have matched (The)
    • False negatives (Type II)

Errors cont.

• In NLP we are always dealing with these kinds of errors.
• Reducing the error rate for an application often involves two antagonistic efforts:
  • Increasing accuracy or precision (minimizing false positives)
  • Increasing coverage or recall (minimizing false negatives).
Summary

- Regular expressions play a surprisingly large role
  - Sophisticated sequences of regular expressions are often the first model for any text processing text
  - I am assuming you know, or will learn, in a language of your choice
- For many hard tasks, we use machine learning classifiers
  - But regular expressions are used as features in the classifiers
  - Can be very useful in capturing generalizations

In Class Exercise

Cars that drive themselves — even parking at their destination — could be ready for sale within a decade, General Motors Corp. executives say. 'This is not science fiction,' Larry Burns, GM's vice president for research and development, said in a recent interview. GM plans to use an inexpensive computer chip and an antenna to link vehicles equipped with driverless technologies. The first use likely would be on highways; people would have the option to choose a driverless mode while they still would control the vehicle on local streets, Burns said.

<table>
<thead>
<tr>
<th>^[^A-Z]</th>
<th>No upper case letters in the line (2, 6, 8, 9)</th>
</tr>
</thead>
</table>

destination - could be ready for sale within a decade, fiction,' Larry Burns, GM's vice president for research and development, said in a recent interview. GM plans to use an inexpensive computer chip and an antenna to link vehicles equipped with driverless technologies. The first use likely would be on highways; people would have the option to choose a driverless mode while they still would control the vehicle on local streets, Burns said.
Basic Text Processing

Word tokenization

Text Normalization

• Every NLP task needs to do text normalization:
  1. Segmenting/tokenizing words in running text
  2. Normalizing word formats
  3. Segmenting sentences in running text
How many words?

• I do uh main- mainly business data processing
  • Fragments, filled pauses
• Terminology
  • **Lemma**: same stem, part of speech, rough word sense
    • *cat* and *cats* = same lemma
  • **Wordform**: the full inflected surface form
    • *cat* and *cats* = different wordforms

How many words?

they lay back on the San Francisco grass and looked at the stars and their

• **Type**: an element of the vocabulary.
• **Token**: an instance of that type in running text.
• How many?
  • 15 tokens (or 14)
  • 13 types (or 12) (or 11?)
How many words?

\[ N = \text{number of tokens} \]
\[ V = \text{vocabulary} = \text{set of types} \]

\(|V|\) is the size of the vocabulary

|                        | Tokens = N  | Types = |\(|V|\)| |
|------------------------|-------------|---------|--------|
| Switchboard phone      | 2.4 million | 20 thousand |
| conversations          |             |         |
| Shakespeare            | 884,000     | 31 thousand |
| Google N-grams         | 1 trillion  | 13 million |

Issues in Tokenization

- Finland’s capital
- what’re, I’m, isn’t
- state-of-the-art
- San Francisco
Issues in Tokenization

- Finland’s capital → Finland Finlands Finland’s ?
- what’re, I’m, isn’t → What are, I am, is not
- state-of-the-art → state of the art ?
- San Francisco → one token or two?

Tokenization: language issues

- Chinese and Japanese no spaces between words:
  - 莎拉波娃 现在 居住 在 美国 东南部 的 佛罗里达。
  - 莎拉波娃 现在 居住 在 美国 东南部 的 佛罗里达
  - Sharapova now lives in US southeastern Florida
Basic Text Processing

Word Normalization and Stemming

Normalization

• Need to “normalize” terms
  • Information Retrieval: indexed text & query terms must have same form.
    • We want to match U.S.A. and USA
• We implicitly define equivalence classes of terms
  • e.g., deleting periods in a term
• Alternative: asymmetric expansion:
  • Enter: windows Search: Windows, windows, window
• Potentially more powerful, but less efficient
Case folding

- Applications like IR: reduce all letters to lower case
  - Since users tend to use lower case
  - Possible exception: upper case in mid-sentence?
    - e.g., General Motors
    - Fed vs. fed
    - SAIL vs. sail
- For sentiment analysis, MT, Information extraction
  - Case is helpful (US versus us is important)

Lemmatization

- Reduce inflections or variant forms to base form
  - am, are, is → be
  - car, cars, car's, cars' → car
  - the boy's cars are different colors → the boy car be different color
- Lemmatization: have to find correct dictionary headword form
Morphology

- **Morphemes:**
  - The small meaningful units that make up words
- **Stems:** The core meaning-bearing units
- **Affixes:** Bits and pieces that adhere to stems
  - Often with grammatical functions

Stemming

- Reduce terms to their stems in information retrieval
- *Stemming* is crude chopping of affixes
  - language dependent
  - e.g., *automate(s), automatic, automation* all reduced to *automat*.

*for example compressed and compression are both accepted as equivalent to compress.*
*for example compress and compress ar both accept as equival to compress*
Porter’s algorithm
The most common English stemmer

Step 1a
- sses → ss  caresses → caress
- ies → i  ponies → poni
- ss → ss  caress → caress
- s → ø  cats → cat

Step 1b
- (*v*)ing → ø  walking → walk
-  (*v*)ed → ø  plastered → plaster

Step 2
- ational → ate  relational → relate
- izer → ize  digitizer → digitize
- ator → ate  operator → operate

Step 3
- al → ø  revival → reviv
- able → ø  adjustable → adjust
- ate → ø  activate → activ

Viewing morphology in a corpus
Why only strip –ing if there is a vowel?

- (*v*)ing → ø  walking → walk
-  sing → sing
Viewing morphology in a corpus
Why only strip –ing if there is a vowel?

(*v*)ing → ø  walking → walk
    sing → sing

```
tr -sc 'A-Za-z' '
' < shakes.txt | grep 'ing$' | sort | uniq -c | sort -nr
1312 King
 548 being
 541 nothing
 388 king
 375 bring
 358 thing
 307 ring
 152 something
 145 coming
 130 morning
 122 having
 120 living
 117 loving
 116 Being
 102 going

tr -sc 'A-Za-z' '
' < shakes.txt | grep '[aeiou].*ing$' | sort | uniq -c | sort -nr
```

Sentence Segmentation

- !, ? are relatively unambiguous
- Period “.” is quite ambiguous
  - Sentence boundary
  - Abbreviations like Inc. or Dr.
  - Numbers like .02% or 4.3
- Build a binary classifier
  - Looks at a “.”
  - Decides EndOfSentence/NotEndOfSentence
  - Classifiers: hand-written rules, regular expressions, or machine-learning
Minimum Edit Distance

Definition and Computation

How similar are two strings?

• Spell correction
  • The user typed “graffe”
  Which is closest?
  • graf
  • graft
  • grail
  • giraffe

• Computational Biology
  • Align two sequences of nucleotides
    AGGCTATCACCTGACCTCCAGCCGATGCC
    TAGCTATCACGACCGCGGTCGATTTGCCC
    --AGCTATCAC--GACC--GGCT
    --TAGCTATCAC--GACC--GGCT
    ----TGCCC--
    --TTGCCC

• Also for Machine Translation, Information Extraction, Speech Recognition
Edit Distance

- The minimum edit distance between two strings
- Is the minimum number of editing operations
  - Insertion
  - Deletion
  - Substitution
- Needed to transform one into the other

Minimum Edit Distance

- Two strings and their alignment:

  INTE*NTION
  | | | | | | | | | |
  *EXECUTION
Minimum Edit Distance

INTEGRATION

EXECUTION
dissis

• If each operation has cost of 1
  • Distance between these is 5
• If substitutions cost 2 (Levenshtein)
  • Distance between them is 8

Other uses of Edit Distance in NLP

• Evaluating Machine Translation and speech recognition
  R Spokesman confirms senior government adviser was shot
  H Spokesman said the senior adviser was shot dead
  S I D I

• Named Entity Extraction and Entity Coreference
  • IBM Inc. announced today
  • IBM profits
  • Stanford President John Hennessy announced yesterday
  • for Stanford University President John Hennessy
ArgRewrite (Profs. Litman & Hwa; subjects needed!)

What’s different about this alignment from prior examples?

<table>
<thead>
<tr>
<th>Draft 1</th>
<th>Draft 2</th>
<th>2-2, Modify, Thesis</th>
<th>3-Null, Delete, Reasoning</th>
<th>Null-3, Add, Reasoning</th>
</tr>
</thead>
<tbody>
<tr>
<td>The next level is the level for angry people.</td>
<td>The next level is the level for angry people.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Saddam Hussein can be put in this circle.</td>
<td>Osama Bin Laden can be put in this circle.</td>
<td>2-2, Modify, Thesis</td>
<td>3-Null, Delete, Reasoning</td>
<td>Null-3, Add, Reasoning</td>
</tr>
<tr>
<td>He is a ruthless dictator and killed many people.</td>
<td>He is a terrorist and killed innocent people.</td>
<td>2-2, Modify, Thesis</td>
<td>3-Null, Delete, Reasoning</td>
<td>Null-3, Add, Reasoning</td>
</tr>
<tr>
<td>He deserves to be in the level of Hell.</td>
<td>He deserves to be in the level of Hell.</td>
<td>2-2, Modify, Thesis</td>
<td>3-Null, Delete, Reasoning</td>
<td>Null-3, Add, Reasoning</td>
</tr>
</tbody>
</table>

How to find the Min Edit Distance?

• Searching (*like in CS1571*) for a path (sequence of edits) from the start string to the final string:
  • **Initial state**: the word we’re transforming
  • **Operators**: insert, delete, substitute
  • **Goal state**: the word we’re trying to get to
  • **Path cost**: what we want to minimize: the number of edits

```
intention
   Del

intention
   Ins
       Sub

intention
```
Minimum Edit as Search

- But the space of all edit sequences is huge!
  - We can’t afford to navigate naïvely
  - Lots of distinct paths wind up at the same state.
    - We don’t have to keep track of all of them
    - Just the shortest path to each of those revisited states.

Defining Min Edit Distance

- For two strings
  - X of length n
  - Y of length m
- We define D(i,j)
  - the edit distance between X[1..i] and Y[1..j]
    - i.e., the first i characters of X and the first j characters of Y
  - The edit distance between X and Y is thus D(n,m)
Dynamic Programming for Minimum Edit Distance

- **Dynamic programming**: A tabular computation of $D(n,m)$
- Solving problems by combining solutions to subproblems.
- **Bottom-up**
  - We compute $D(i,j)$ for small $i,j$
  - And compute larger $D(i,j)$ based on previously computed smaller values
  - i.e., compute $D(i,j)$ for all $i$ ($0 < i < n$) and $j$ ($0 < j < m$)

Defining Min Edit Distance (Levenshtein)

- **Initialization**
  $D(i,0) = i$
  $D(0,j) = j$

- **Recurrence Relation**:  
  For each $i = 1...M$  
  For each $j = 1...N$
  
  $D(i,j) = \min \left\{ \begin{array}{l}
  D(i-1,j) + 1 \\
  D(i,j-1) + 1 \\
  D(i-1,j-1) + 2; \quad \text{if } X(i) \neq Y(j) \\
  0; \quad \text{if } X(i) = Y(j)
  \end{array} \right.$

- **Termination**:  
  $D(N,M)$ is distance
### The Edit Distance Table

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>O</th>
<th>I</th>
<th>T</th>
<th>N</th>
<th>E</th>
<th>T</th>
<th>N</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
</tr>
</tbody>
</table>

# EXECUTION
### Edit Distance

\[
D(i, j) = \min \begin{cases} 
D(i-1, j) + 1 \\
D(i, j-1) + 1 \\
D(i-1, j-1) + 2; \text{ if } S_1(i) \neq S_2(j) \\
0; \text{ if } S_1(i) = S_2(j)
\end{cases}
\]

<table>
<thead>
<tr>
<th>N</th>
<th>9</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>O</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>#</td>
<td>E</td>
<td>X</td>
<td>E</td>
<td>C</td>
<td>U</td>
<td>T</td>
<td>I</td>
<td>O</td>
<td>N</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### The Edit Distance Table

<table>
<thead>
<tr>
<th>N</th>
<th>9</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>11</th>
<th>10</th>
<th>9</th>
<th>8</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>O</td>
<td>8</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>I</td>
<td>7</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>7</td>
<td>8</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>#</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>#</td>
<td>E</td>
<td>X</td>
<td>E</td>
<td>C</td>
<td>U</td>
<td>T</td>
<td>I</td>
<td>O</td>
<td>N</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Computing alignments

- Edit distance isn’t sufficient
  - We often need to align each character of the two strings to each other
- We do this by keeping a “backtrace”
- Every time we enter a cell, remember where we came from
- When we reach the end,
  - Trace back the path from the upper right corner to read off the alignment

In Class Exercise

- Using 1-insertion, 1-deletion, 2-substitution costs, compute the minimum distance between drive (left column) and brief (bottom row)
Weighted Edit Distance

- Why would we add weights to the computation?
  - Spell Correction: some letters are more likely to be mistyped than others
  - Biology: certain kinds of deletions or insertions are more likely than others
- Also other variants on basic algorithm (e.g., global context)

Confusion matrix for spelling errors

|   | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
| a | 0 | 0 | 7 | 0 | 5 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 24 |
| b | 0 | 0 | 9 | 9 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| c | 1 | 10 | 13 | 0 | 12 | 0 | 5 | 5 | 0 | 0 | 0 | 2 | 3 | 7 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| d | 0 | 3 | 11 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 5 | 5 | 0 | 0 | 0 | 14 | 12 | 6 | 15 | 0 | 1 | 0 | 0 | 0 |
| e | 0 | 15 | 0 | 3 | 1 | 0 | 5 | 2 | 0 | 0 | 0 | 0 | 0 | 3 | 4 | 1 | 0 | 0 | 0 | 6 | 4 | 12 | 0 | 0 | 0 | 2 | 0 | 0 | 0 |
| f | 4 | 1 | 11 | 11 | 9 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | 3 | 0 | 0 | 2 | 1 | 3 | 5 | 15 | 21 | 0 | 0 | 1 | 0 | 0 | 0 |
| g | 1 | 8 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 12 | 14 | 2 | 3 | 0 | 3 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| h | 103 | 0 | 0 | 0 | 146 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 6 | 0 | 0 | 0 | 49 | 0 | 0 | 0 | 0 | 2 | 1 | 47 | 0 | 2 | 1 | 15 | 0 |
| i | 0 | 1 | 1 | 9 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| j | 1 | 2 | 8 | 4 | 1 | 1 | 2 | 5 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| k | 2 | 10 | 1 | 4 | 0 | 4 | 5 | 6 | 13 | 0 | 1 | 0 | 0 | 0 | 14 | 2 | 5 | 0 | 11 | 10 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| l | 1 | 2 | 7 | 8 | 0 | 2 | 0 | 6 | 0 | 0 | 0 | 4 | 0 | 0 | 180 | 0 | 6 | 0 | 0 | 9 | 15 | 13 | 3 | 2 | 2 | 3 | 0 |
| m | 2 | 7 | 6 | 5 | 3 | 0 | 1 | 19 | 1 | 0 | 4 | 55 | 78 | 0 | 0 | 7 | 0 | 28 | 5 | 7 | 0 | 0 | 1 | 2 | 0 | 2 | 0 | 2 |
| n | 91 | 1 | 1 | 3 | 116 | 0 | 0 | 0 | 25 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 14 | 0 | 2 | 4 | 14 | 39 | 0 | 0 | 0 | 18 | 0 |
| o | 0 | 1 | 1 | 2 | 0 | 6 | 5 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 3 | 6 | 0 | 4 | 1 | 0 | 0 | 0 | 0 | 0 |
| p | 0 | 0 | 1 | 0 | 0 | 0 | 27 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| q | 0 | 14 | 0 | 30 | 12 | 2 | 2 | 7 | 8 | 2 | 0 | 5 | 8 | 4 | 20 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 12 | 22 | 4 | 0 | 0 | 1 | 0 | 0 |
| r | 11 | 8 | 27 | 33 | 35 | 4 | 0 | 1 | 1 | 0 | 27 | 0 | 6 | 1 | 7 | 0 | 0 | 0 | 1 | 3 | 14 | 0 | 0 | 5 | 5 | 20 | 1 |
| s | 3 | 4 | 9 | 42 | 7 | 5 | 19 | 5 | 0 | 1 | 14 | 9 | 5 | 5 | 0 | 11 | 37 | 0 | 0 | 0 | 2 | 19 | 0 | 7 | 6 | 0 | 0 | 0 | 0 |
| t | 20 | 0 | 0 | 0 | 44 | 0 | 6 | 0 | 64 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 43 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 8 | 6 | 0 | 0 | 0 |
| u | 0 | 0 | 7 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| v | 2 | 2 | 1 | 0 | 1 | 0 | 0 | 0 | 2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| w | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| x | 0 | 0 | 2 | 0 | 15 | 0 | 1 | 7 | 15 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 6 | 1 | 0 | 0 | 0 | 7 | 36 | 8 | 5 | 0 | 0 | 1 | 0 |
| y | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7 | 5 | 0 | 0 | 0 | 0 | 2 | 21 | 3 | 0 | 0 | 0 | 0 | 3 | 0 |
| z | 24 |