Translation Memory Engines: A Look under the Hood and Road Test

Timothy Baldwin
tbaldwin@csli.stanford.edu

CSLI CENTER FOR THE STUDY OF LANGUAGE AND INFORMATION

http://lingo.stanford.edu/pubs/tbaldwin/ijet2004.pdf

What this Talk isn't

- A commercial endorsement of translation memories or any translation memory product
- A demonstration of any particular translation memory product
- A discussion of integration issues with everyone's favourite word processing application

What this Talk is

- The ramblings of a researcher who stumbled upon translation memories through occasional translation work
- (Hopefully) a step towards the demystification of translation memories
- A bit of a play around with a technology of practical utility

Structure of this Talk

- **Part I:** A look under the hood of translation memories

Coffee break (5 minutes)

- **Part II:** Road test of Japanese–English translation retrieval

PART I: A Look under the Hood of Translation Memories

The Reality of Translation

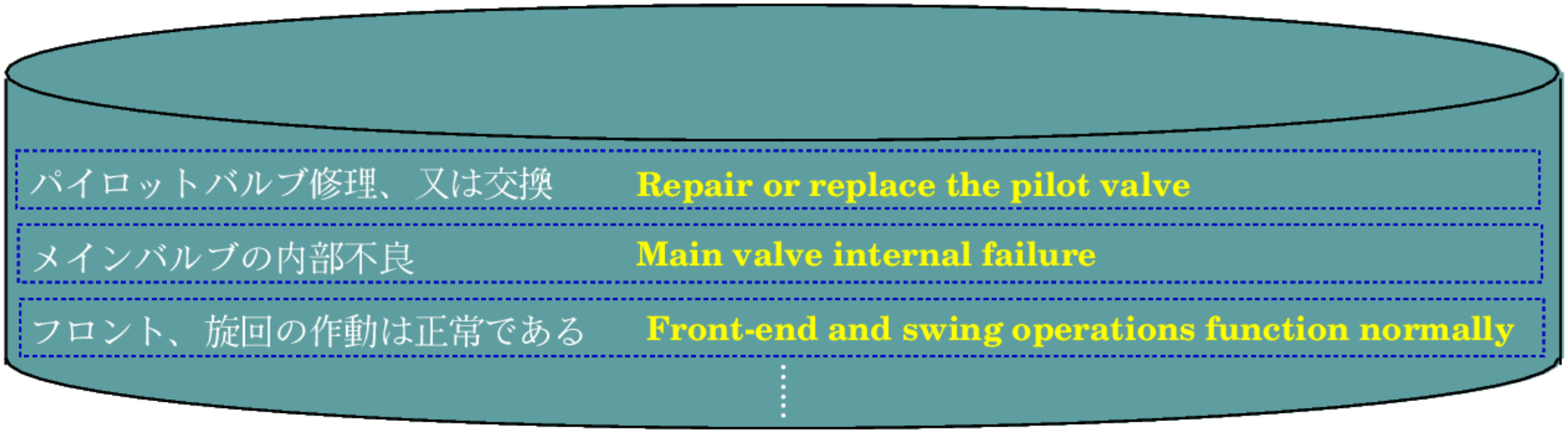
- The staple diet of translation work is technical materials (controlled languages = contracts, user manuals, field reports, etc.) NOT literary prose
- Technical translation calls for in- and inter-document **consistency** and translation accuracy
- It is time-consuming and expensive to train technical translators
- Technical translation is repetitive!

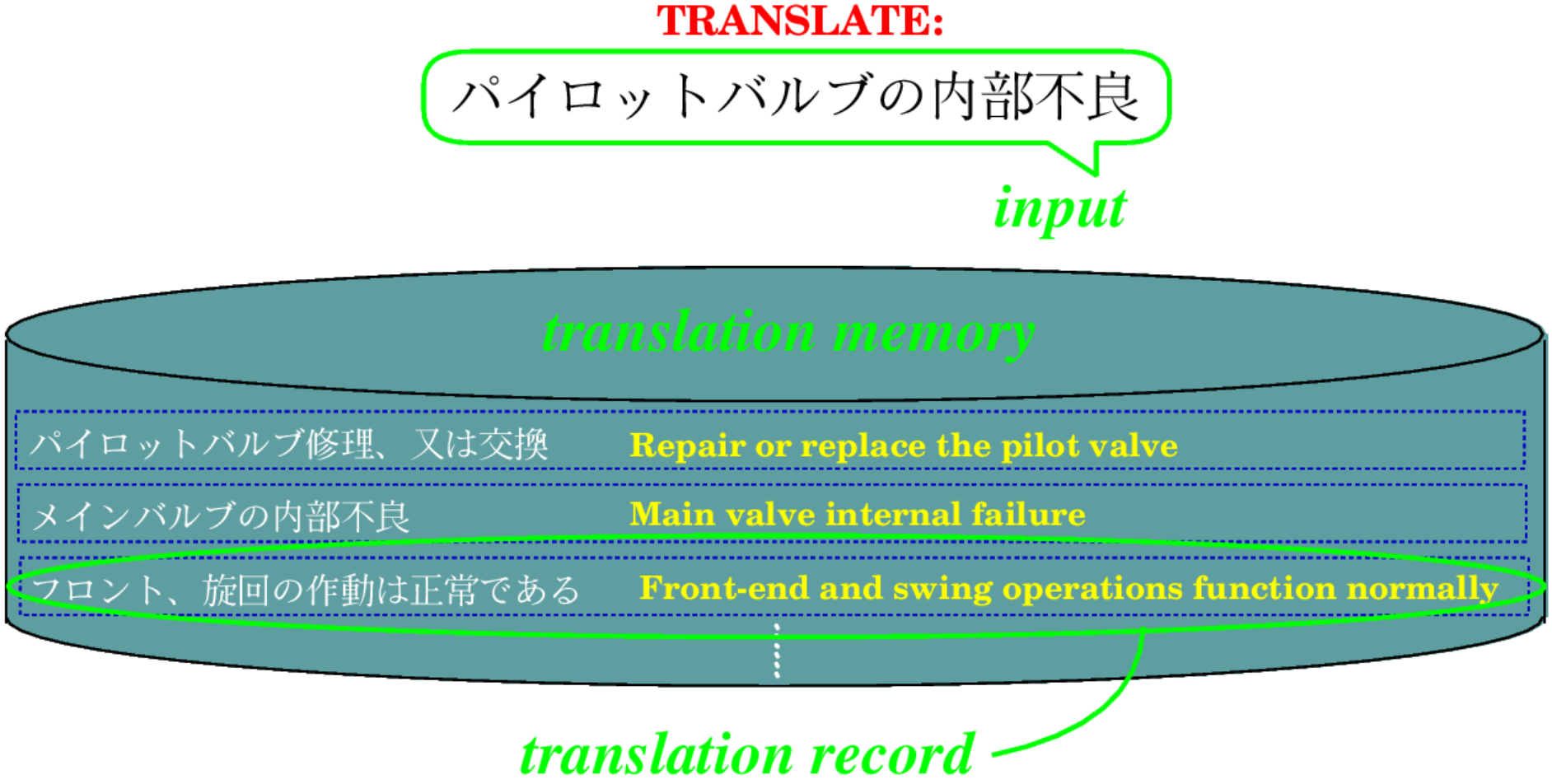
Enter the Computer!

- Computers are consistent (although possibly consistently $wrong$!)
- Computers have an infinite boredom tolerance
- Computers are good at performing tasks they know how to do (e.g. translate a given string based on a human translation)
- BUT computers (machine translation systems) are temperamental at best at translating novel strings

TRANSLATE:

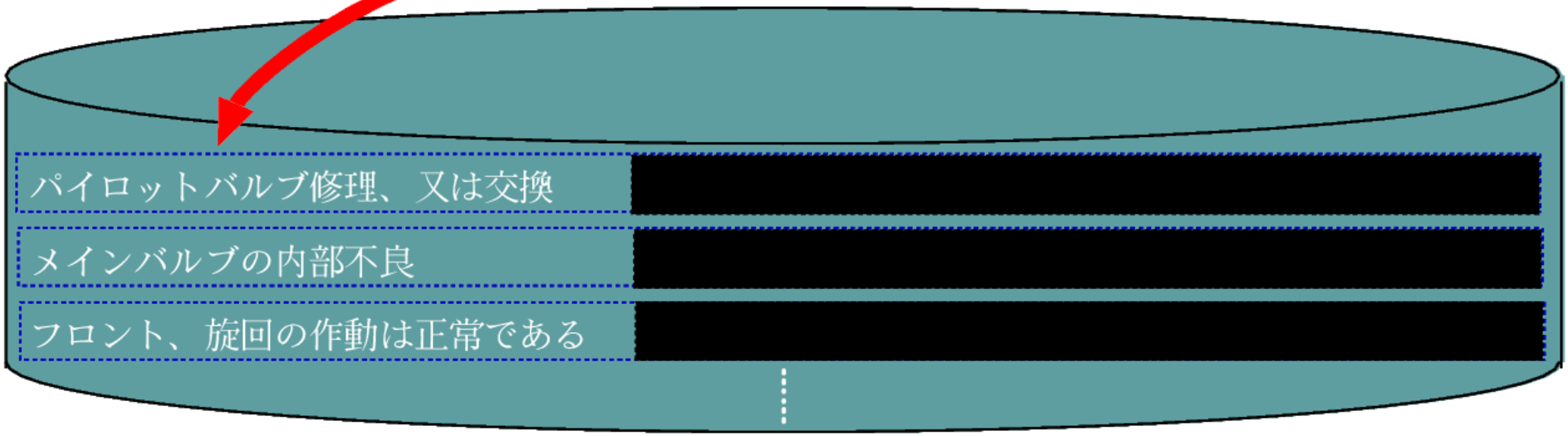
パイロットバルブの内部不良




TRANSLATE:
パイロットバルブの内部不良
input
translation memory
パイロットバルブ修理、又は交換
Repair or replace the pilot valve
メインバルブの内部不良
Main valve internal failure
フロント、旋回の作動は正常である
Front-end and swing operations function normally
translation record

TRANSLATE:

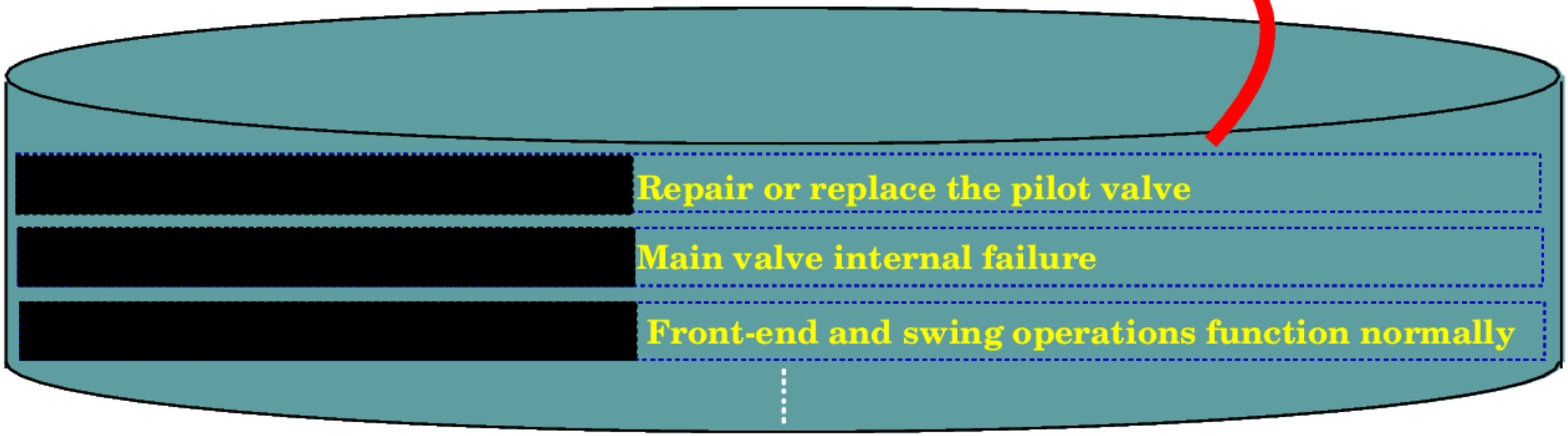
パイロットバルブの内部不良



TRANSLATE:

パイロットバルブの内部不良

{ Repair or replace the pilot valve
Main valve internal failure }



Key Terms

- **Translation record:** source language (L1) string coupled with its target language (L2) translation
- **Translation memory (TM):** database of translation records
- **Translation retrieval (TR):** the process of retrieving translation(s) from the translation memory based on L1 similarity with the input

Human–computer Symbiosis (1)

- The role of the computer:
  - ⋆ translate previously-seen inputs (by way of previously-generated translations)
  - ⋆ suggest translations for inputs similar to previously-seen inputs
  - ⋆ check on the consistency of the translation (based on past translation data)
  - ⋆ streamline the translation process so as to minimise translator effort

Human–computer Symbiosis (2)

- The role of the human:
  - ⋆ concentrate on novel data
  - ⋆ post-edit any computer-generated translations
  - ⋆ post-edit the final document to ensure readability, factual accuracy

Translation Memory Feedback Loop

1. TM system suggests translation candidates for given input based on best L1 matches in translation memory

2. User fashions translation for current input with or without a translation candidate from the TM

3. TM system feeds the final translation (and original input) back into the translation memory for use in subsequent retrievals

Key Assumptions

- **Uniqueness of translation:** a given L1 string has a unique (or limited number of) optimal translation in the given domain
- **Lexico-semantic similarity:** for strings S and T, $sim_{lex}(S,T) \propto sim_{sem}(S,T)$
- **Cross-lingual similarity:** for translation records $\langle w, x \rangle$ and $\langle y, z \rangle$, $sim_{SL}(w,y) \propto sim_{TL}(x,z)$

Optional Assumption

- **Discourse independence:** an optimal document-level translation can be acquired by concatenating L2 units from the translation memory in the same order as the original L1 units

HCI-related Desiderata for TM Systems

- Speed: the user shouldn't be kept waiting for TM output
- Invisibility: the only interaction the user should have with the TM system is in choosing whether or not to recycle previous translation data in translating a given string
- Seamlessness: the user should not have to make any sacrifices in terms of text processing functionality in order to use a TM system

Orthogonal Parameters in TR

- **Segment granularity**: unit character vs. word segments (character- vs. word-based indexing)
- **Segment order**: segment order-oblivion (bag-of-words approach) vs. segment order-sensitivity
- **Segment contiguity**: segment contiguity oblivion vs. segment contiguity awareness

Example

- Translation memory:
  - (1a) 夏の雨 [$natu\ no\ ame$] "summer rain"
  - (1b) 雨の夏 [$ame\ no\ natu$] "a rainy summer"
  - (1c) 雨の冬 [$ame\ no\ huyu$] "a rainy winter"
  - (1d) 真冬の雨 [$ma\ huyu\ no\ ame$] "mid-winter rain"
  - (1e) にわか雨 [$niwakaame$] "rain shower"
- Translate:
  - (2) 冬の雨 [$huyu\ no\ ame$] "winter rain"

Segment Granularity

- What are the basic units/segments in retrieval?
  - ⋆ Characters:

    (1d) 真·冬·の ·雨
    (1e) に·わ·か·雨

  - ⋆ Words (morphemes):

    (1d) 真冬·の ·雨
    (1e) にわか雨

Segmentation Modules Used

- ChaSen (NAIST)
- JUMAN (Kyoto Uni.)
- ALTJAWS (NTT CS Labs.)

Segment Order

- Does the order of the segments matter?
  - ⋆ Segment order-oblivious (bag of words) methods :
    (1c) 雨·の ·冬 ≡ 冬·の ·雨 ≡ 冬·雨·の ...

    ⋮

    Venetian blind ≡ blind Venetian (!!!)
  - ⋆ Segment order-sensitive methods:
    (1c) 雨の冬 ≠ 冬の雨 ≠ 冬雨の ...

    ⋮

    Venetian blind ≠ blind Venetian

Segment Contiguity

- Is segment contiguity significant?

sim(雨の多い冬, 雨の冬) > sim(雨マークの多い冬, 雨の冬)

sim(強い雨の冬, 雨の冬) > sim(雨の強い冬, 雨の冬) ??

String Comparison Methods

- **Bag-of-words methods**:
  - ⋆ Vector space model
  - ⋆ Token intersection
- **Segment order-sensitive methods**:
  - ⋆ 3-operation edit distance
  - ⋆ 3-operation edit similarity
  - ⋆ Weighted sequential correspondence

Vector Space Model

- For vectors $\vec{S}$ and $\vec{T}$ of segment types in strings S and T, respectively:

$$\cos(\vec{S},\vec{T}) = \frac{\sum_j s_j t_j}{\sqrt{\sum_j s_j^2}\sqrt{\sum_j t_j^2}}$$

- Segment order-insensitive similarity

Example

- Input:

  (2) 冬·の·雨 $= [1, 1, 1, 0, 0, 0, 0]$

- Translation memory:

  (1a) 夏·の·雨 $= [0, 1, 1, 1, 0, 0, 0]$; $sim((1a),(2)) = \frac{2}{3}$
  (1b) 雨·の·夏 $= [0, 1, 1, 1, 0, 0, 0]$; $sim((1b),(2)) = \frac{2}{3}$
  (1c) 雨·の·冬 $= [1, 1, 1, 0, 0, 0, 0]$; $sim((1c),(2)) = \frac{3}{3}$
  (1d) 真·冬·の·雨 $= [1, 1, 1, 1, 0, 0, 0]$; $sim((1d),(2)) = \frac{3}{2\sqrt{3}}$
  (1e) に·わ·か·雨 $= [0, 0, 1, 0, 1, 1, 1]$; $sim((1e),(2)) = \frac{1}{2\sqrt{3}}$

Token Intersection (=Dice's Coefficient)

- For strings S and T of (weighted) segment length $length(S)$ and $length(T)$, respectively:

$$tint(S,T) = \frac{2 \times \sum_{e \in S,T} \min\left(freq_S(e), freq_T(e)\right)}{length(S) + length(T)}$$

- Segment order-insensitive similarity

Example

- Input:

  (2) 冬·の·雨

- Translation memory:

  (1a) 夏·の·雨 ; sim((1a),(2)) = $\frac{4}{6}$
  (1b) 雨·の·夏 ; sim((1b),(2)) = $\frac{4}{6}$
  (1c) 雨·の·冬 ; sim((1c),(2)) = $\frac{6}{6}$
  (1d) 真·冬·の·雨 ; sim((1d),(2)) = $\frac{6}{7}$
  (1e) に·わ·か·雨 ; sim((1e),(2)) = $\frac{2}{7}$

3-operation Edit Distance

- Distance between strings in terms of the number of segment insertions and deletions required to transform one into the other
- Segment order-sensitive **distance**

$$D_{3op}(S,T) \;=\; d_3(m,n)$$

$$d_3(i,j) = \begin{cases} 0 & if\ i = 0 \wedge j = 0 \\ d_3(0,j-1)+1 & if\ i = 0 \wedge j \neq 0 \\ d_3(i-1,0)+1 & if\ i \neq 0 \wedge j = 0 \\ \min \begin{pmatrix} d_3(i-1,j)+1, \\ d_3(i,j-1)+1, \\ m_3(i,j) \end{pmatrix} & otherwise \end{cases}$$

$$m_3(i,j) = \begin{cases} d_3(i-1,j-1) & if\ s_i = s_j \\ \infty & otherwise \end{cases}$$

Example

- Input:

  (2) 冬·の·雨

- Translation memory:

  (1a) 夏·の·雨 ; D_{3op}((1a),(2)) = 2
  (1b) 雨·の·夏 ; D_{3op}((1b),(2)) = 4
  (1c) 雨·の·冬 ; D_{3op}((1c),(2)) = 4
  (1d) 真·冬·の·雨 ; D_{3op}((1d),(2)) = 1
  (1e) に·わ·か·雨 ; D_{3op}((1e),(2)) = 6

3-operation Edit Similarity

- Normalise edit distance $S_{3op}(S,T)$ according to weighted lengths of the strings S and T:

$$S_{3op}(S,T) = 1 - \frac{D_{3op}(S,T)}{length(S) + length(T)}$$

- Segment-order sensitive similarity
- Best match for (1d): $S_{3op}((2),(1d)) = \frac{6}{7}$

Example

- Input:

  (2) 冬·の·雨

- Translation memory:

  (1a) 夏·の·雨 ; $S_{3op}((1a),(2)) = \frac{2}{3}$
  (1b) 雨·の·夏 ; $S_{3op}((1b),(2)) = \frac{1}{3}$
  (1c) 雨·の·冬 ; $S_{3op}((1c),(2)) = \frac{1}{3}$
  (1d) 真·冬·の·雨 ; $S_{3op}((1d),(2)) = \frac{6}{7}$
  (1e) に·わ·か·雨 ; $S_{3op}((1e),(2)) = \frac{1}{7}$

Weighted Sequential Correspondence

- Degree of contiguous substring similarity between strings, weighted according to the contiguity of segment matches
- Assume $Max = 3$
- Segment order/contiguity-sensitive similarity
- Best match for (1d): $wseq^*((2), (1d)) = \frac{2 \times 6}{6+9} = \frac{4}{5}$

$$S_w(S,T) \;=\; s(m,n)$$

$$s(i,j) = \begin{cases} 0 & if\ i = 0 \vee j = 0 \\ \max \left(\begin{array}{l} s(i-1,j), \\ s(i,j-1), \\ s(i-1,j-1) + m_w(i,j) \end{array} \right) & otherwise \end{cases}$$

$$m_w(i,j) = \begin{cases} cm(i,j) & if\ s_i = s_j \\ 0 & otherwise \end{cases}$$

$$cm(i,j) = \begin{cases} 0 & if\ i = 0 \vee j = 0 \vee s_i \neq t_j \\ \min(Max, cm(i-1,j-1)+1) & otherwise \end{cases}$$

$$wseq^*(S,T) = \frac{2 \times S_w(S,T)}{S_w(S,S) + S_w(T,T)}$$

Example

- Input:

  (2) 冬·の·雨

- Translation memory:

  (1a) 夏·の·雨 ; $wseq^*$((1a),(2)) = $\frac{6}{12}$
  (1b) 雨·の·夏 ; $wseq^*$((1b),(2)) = $\frac{2}{12}$
  (1c) 雨·の·冬 ; $wseq^*$((1c),(2)) = $\frac{2}{12}$
  (1d) 真·冬·の·雨 ; $wseq^*$((1d),(2)) = $\frac{12}{15}$
  (1e) に·わ·か·雨 ; $wseq^*$((1e),(2)) = $\frac{2}{15}$

N-gram Models

- Possibility of complementing string comparison methods with model of local segment contiguity (which preserves basic segment order):
  - ⋆ **unigrams**: バルブ = バ·ル·ブ
  - ⋆ **bigrams**: バルブ = バル·ルブ
  - ⋆ **mixed unigrams/bigrams**: バルブ = バ·バル·ル·ルブ·ブ
- Compatible with all models

Predicted Results

- String representation: words better than characters? (YES/NO)
- Segment order matters? (YES/NO)
- Segment contiguity matters? (YES/NO)

Coffee Break (phew!)

Part II: Road Test of Japanese–English Translation Retrieval

Evaluation Datasets

- **Dataset 1**: 3,033 translation records
  - ⋆ from technical field reports
  - ⋆ average translation record length:
    27.7 characters (J)/13.3 words (E)
- **Dataset 2**: 61,236 translation records
  - ⋆ from government white papers
  - ⋆ average translation record length:
    76.3 characters (J)/35.7 words (E)

Dataset 1 – Technical Field Reports

i. モニタから水温センサ間のハーネス断線 = break in the harness from the monitor to the coolant temperature sensor
ii. 又、午前中バッテリ交換した後夕方まで休車させるとバッテリが完全放電してしまった。 = additionally , if battery is replaced in the morning and the machine is left idle for the day , the battery is flat by evening
iii. 点検しハーネスを規定途通りに接続した。 = inspected and corrected the harness connection
iv. fm/amラジオのチャンネルと時計のメモリーがキースイッチoffにすると消える。 = turning the ignition off resets the clock and the channel setting of the am / fm radio

Dataset 2 – Government White Papers

i. 最近10年間の民間企業の研究者数の推移についてみてみると，研究本務者は非常に安定して増加を続けてきている（第2-1-25図）。 = the number of regular researchers has shown a steady increase over the last ten years (figure 2-1-25)

ii. 農林水産省においては，「新需要創出のための生物機能の開発・利用技術の開発に関する総合研究」の一環として，バイオプラスチック・シルクレザー等の生物素材に係る研究開発を実施している。 = the ministry of agriculture forestry and fisheries is promoting research and development on bio-materials such as bio-plastics and silk leather as part of its integrated research program for effective use of biological activities to create new demand

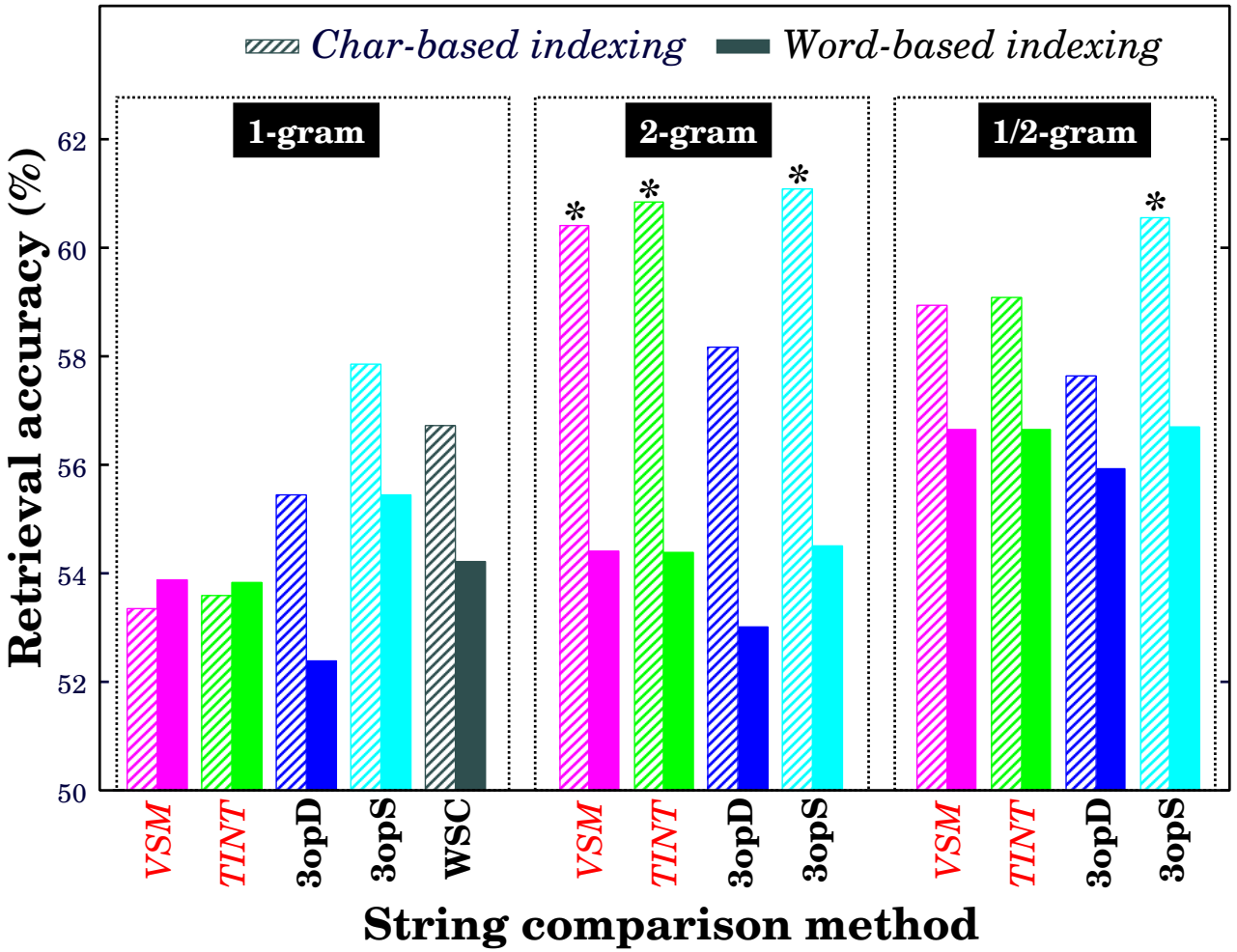
Evaluation Procedure

- Use *10-fold semi-stratified cross-validation* to determine retrieval accuracy
- On each iteration of cross-validation, take the L1 component of the test data as the set of inputs, and the L2 translations as the model translations
- Use an arbitrary string comparison method to determine the set of "optimal translations" for a given input, as the translations in the TM most similar/least displaced

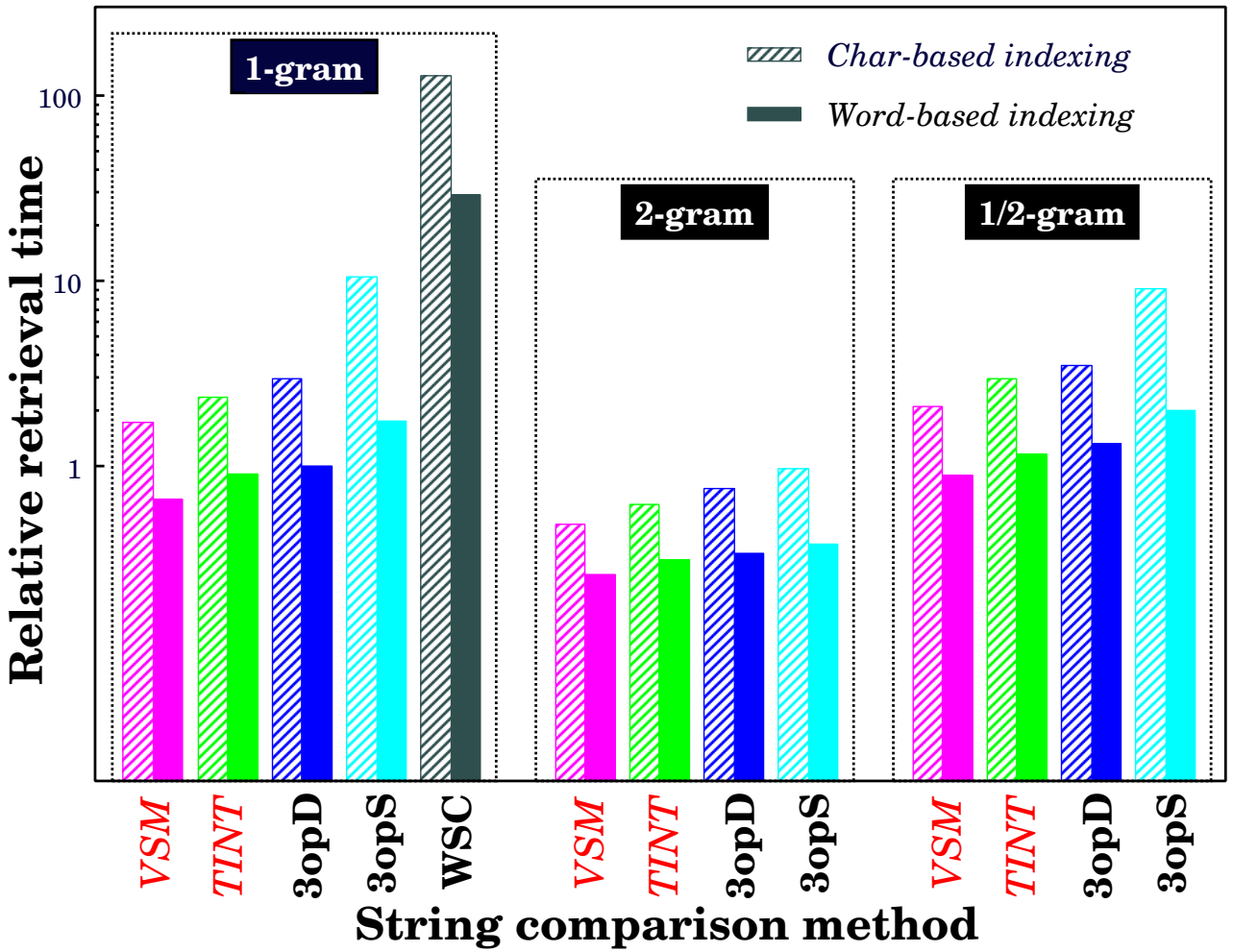
from the model translation

- "Retrieval accuracy" defined as the proportion of inputs for which an optimal translation was retrieved
- Retrieval accuracy averaged over the results from the 3-op edit distance and weighted sequential correspondence methods

Basic Retrieval Accuracies



Basic Retrieval Times



Preliminary Findings (1)

- Char-based indexing is superior to word-based indexing, particularly when combined with bigram and mixed bigram segment contiguity models
- Bag-of-words and segment order-sensitive methods are roughly equivalent in retrieval accuracy, but bag-of-words methods are (much) faster

Preliminary Findings (2)

- Explicit modelling of local segment contiguity does enhance retrieval performance, with bigrams being the best N-gram choice for char-based indexing (both more accurate and faster), and mixed unigrams/bigrams the best choice for word-based indexing
- The best string comparison method is 3-op edit similarity (marginally)

Questions to Come out of this

- Is the superiority of char-based indexing:

(a) a universal trait or particular to the original dataset?
(b) particular to ChaSen or observable for other segmenters also?

- Equivalence of the bag-of-words and segment order-sensitive methods particular to the original dataset?

- How does variation in the size of the dataset affect retrieval performance?

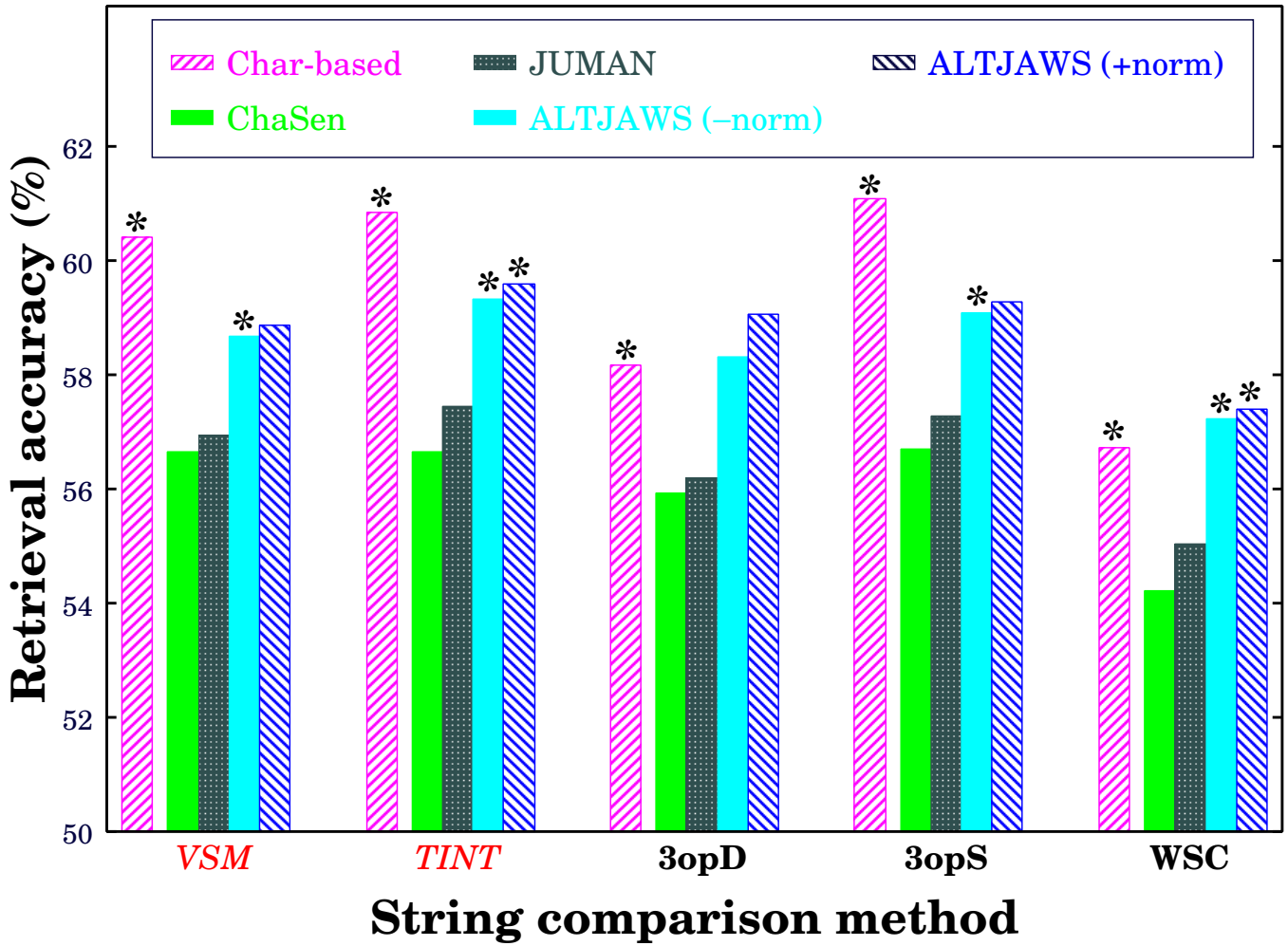
Segmentation and Retrieval Accuracy

- Redo retrieval evaluation with data segmented with:

(a) JUMAN
(b) ALTJAWS (−lexical normalisation)
(c) ALTJAWS (+lexical normalisation)

- Lexical normalisation: convert all morphemes into canonical form, e.g. 行っ|た → 行く|た, 成り行き → 成行き, 充分 → 十分

Results for Different Segmenters



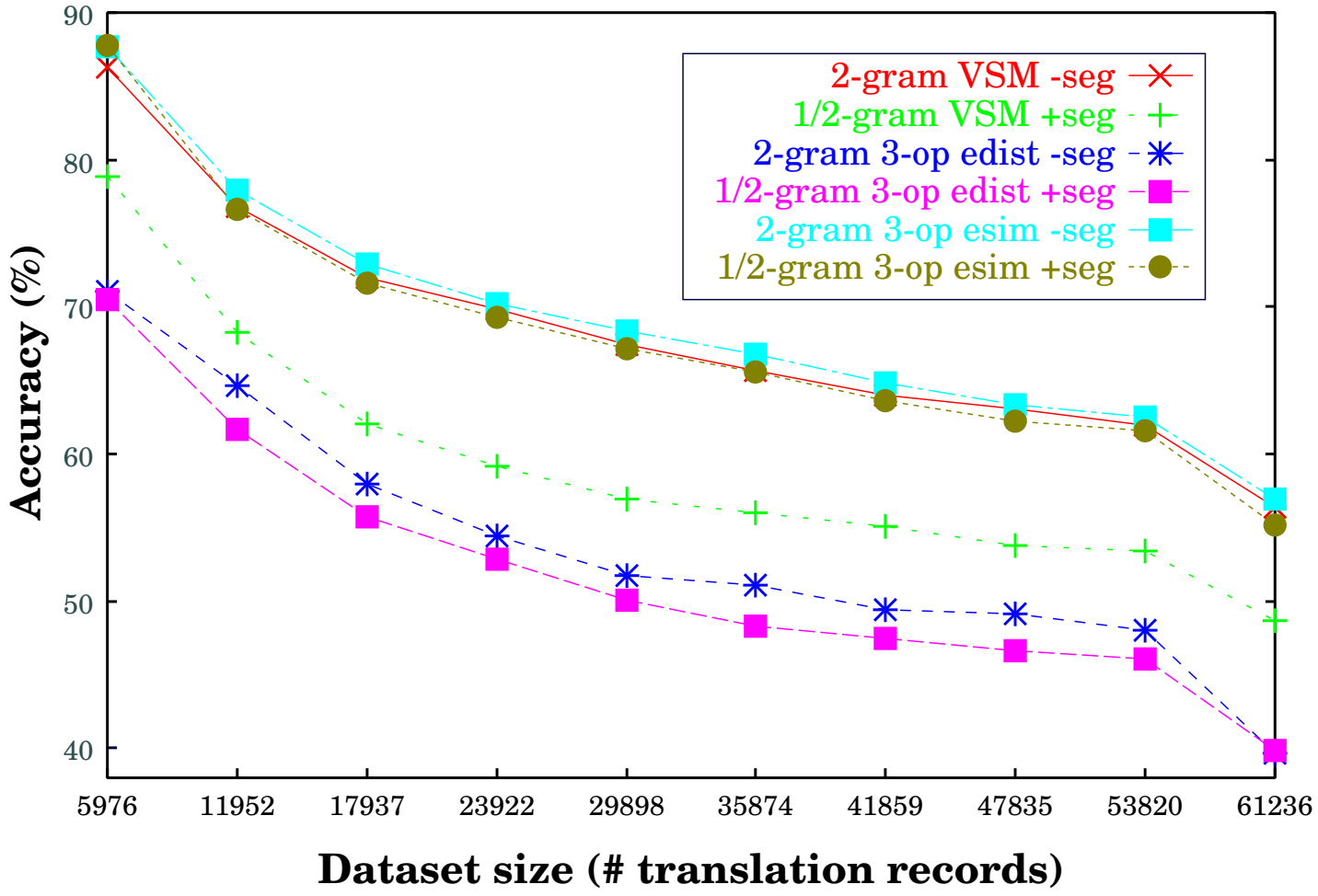
Preliminary Findings

- JUMAN slightly better than ChaSen, ALTJAWS better again
- Lexical normalisation produces only a slight gain (due to the relative infrequency of lexical alternation)
- Char-based indexing still superior in terms of accuracy for best-performing methods (+speed gains)

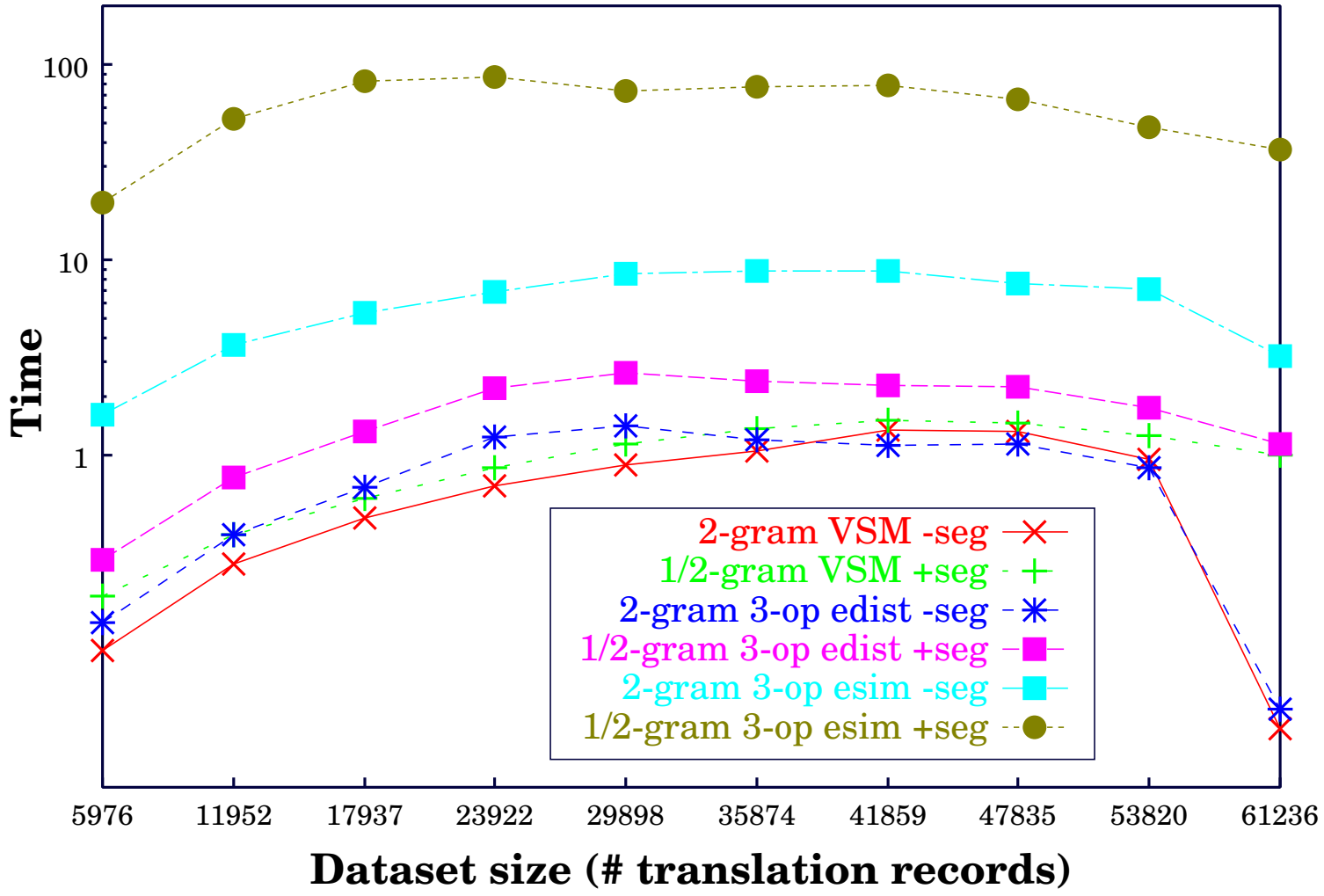
Scalability of Performance

- **Limitation of evaluation to here**: No sense of whether the findings to date are particular to the given dataset or data size
- **Solution**: Take a larger dataset (JEIDA parallel corpus), and evaluate retrieval performance over data increments of increasing size
- Compare the relative accuracy and speed of the vector space model, 3-op edit distance and 3-op edit similarity

Retrieval Accuracy over Different TM Sizes



Retrieval Time over Different TM Sizes



Summary of Results for Performance Scalability

- For all methods tested, the absolute disparity between **character-** and **word-based indexing** was maintained over different data sizes (**character-based indexing** superior to different degrees)
- The best performers were **3-operation edit similarity** and the **vector space model**, under character-based indexing (bag-of-words $\equiv$ segment order-sensitive string comparison)

- **3-operation edit similarity** ran about 10 times slower than the other two methods for char-based indexing, and about 100 times slower for word-based indexing

Qualitative Evaluation

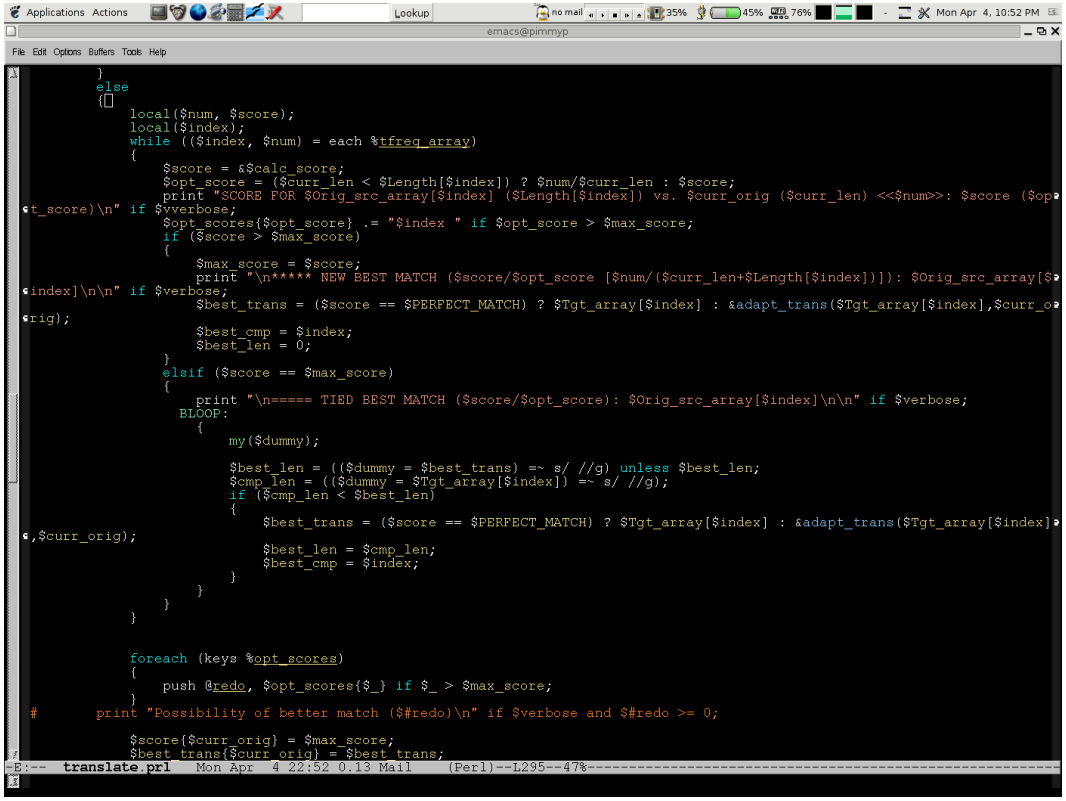
- **Character- vs. word-based indexing**
  - ⋆ longer matching strings scored higher under character-based indexing (picks up on katakana sequences = technical terms)
- **Bag-of-words vs. segment order-sensitive methods**
  - ⋆ if *high* level of segment overlap between strings, small probability of those segments having occurred in different order

- ⋆ if low level of segment overlap between strings, string filtered off by translation utility thresholds

  $\longrightarrow$ very few cases where segment order sensitivity is needed

My Pet Translation Memory System

Snippet of hacky Perl code:



Overall Conclusions (1)

- **Character-based indexing** superior to **word-based indexing**
- **Bag-of-words** roughly equivalent to **segment order-sensitive methods** in terms of accuracy *but much faster*, for the string comparison methods considered (naive but fast is good enough)

Overall Conclusions (2)

- N-gram models of local segment contiguity enhance retrieval performance (both speed and accuracy for char-based indexing)
- The suggested relative running times and accuracies are scalable over variable data sizes

Messages to Take Away

- Translation memories are really very stupid, and the stupider they are, the better they perform
- Translation memories aren't the exclusive domain of private companies/commerical products
- (In some domains at least) translation memories offer real benefits
- Translations memories are fun!