TR－SLT－0038

Corpus Processing for Machine Translation Experiments and Tools

Stephen Nightingale

2003. 4. 30

This report details some experiments in Corpus Processing for Machine Translation, involving public domain and ATR developed tools run in complex sequences. The ultimate objective of SLT Department 4 is Simultaneous Interpretation of News. This work is in support of the Machine Translation component of that goal. The first experiment covers the processes required to accomplish Statistical Machine Translation using the ATR Basic Travel Expressions Corpus. The available News Corpora are not, however, immediately suited to SMT, so subsequent experiments are performed to investigate and extract such parallel, alignable resources as are available. The results of running these experiments are analysed in published papers referenced in the text.The aim of the report is to identify useful tools and configurations for corpus processing and document them.

（株）国際電気通信基礎技術研究所
音声言語コミュニケーション研究所
〒619－0288「けいはんな学研都市」光台二丁目２番地２TEL：0774－95－1301

Advanced Telecommunication Research Institute International
Spoken Language Translation Research Laboratories
2-2-2 Hikaridai "Keihanna Science City" 619-0288,Japan
Telephone:+81-774-95-1301
Fax :+81-774-95-1308

目次

1 Introduction and Project Goals

The ultimate objective of the translation group in Department 4 is to develop a "Simultaneous Interpretation System for Monologue News" (Tanaka et al). Decomposing the problem into its three constituents this involves

- Speech Recognition: recognizing and transcribing the spoken Japanese.
- Machine Translation: translating from Japanese to English.
- Speech Synthesis: uttering the translated English.

Since the objective includes interpretation we would like to translate in the pattern of a human interpreter. Since Machine Translation is implicit in interpretation, we are exploring methods of corpus based MT, using available News Corpora.

The purpose of this document is to summarize some experience in processing bilingual corpora with a view to the Machine Translation component of the Simultaneous Interpretation project. It aims to be a practical guide to processing the available corpora using some useful publicly available tools, and tools developed at ATR for corpus processing. This introduction sets the scene with discussions of three methods of Machine Translation relevant to the inquiry. These are Statistical Machine Translation, Example Based Machine Translation and Translation Memory. It is the interactions between these methods and the available corpora that determine what corpus processing is required.

1.1 Statistical Machine Translation

SMT was inspired by the success of the corpus based Speech Recognition methods developed at IBM [Bahl, Jelinek et al 1983], where Expectation Maximization methods are used on the digitized speech wave and its text transcription to find statistical correspondences between text and sound. [Brown, et al 1993] have developed methods based on the same broad Expectation Maximization scheme to find correspondences between a source text in one language and a target text in another, to train a machine to accept an unseen source sentence and generate a translation using the patterns discovered. The structural complexity of language pairs differs from that of speech and its transcription, so the translation training model differs. In fact there are five interlocking models: Model 1 starts with a bilingual corpus of sentence pairs, and computes word frequency distributions to find the translation probability of Source and Target language word pairs; Model 2 builds on these to compute the probability of word alignments between positions in classes of source and language sentence

pairs, based on their relative lengths; Models 3 and above build on these preliminary alignments to compute word fertility distributions: how many words in the Target map to any given word in the Source, depending on its position. In the EGYPT [Al Onaizan et al 2000] implementation the output of the training phase includes Source and Target vocabularies (word lists and their frequencies), Source|Target translation probability tables, Source|Target alignment tables, and Source|Target word fertility tables.

Translation of an unseen sentence proceeds according to the equation:

1. english = argmax P(Japanese|English) * P(English)

The left-hand side represents the resulting English translation, given a previously unseen Japanese sentence. The right hand side represents the results of training over the corpus and is in two parts:

P(J|E) The translation table outputs represent complex relationships between bilingual word mappings, and over monolingual word mappings. The possible translations constructed from these tables given the unseen input are overgenerated. No thought is given to the grammaticality of the results, which may therefore include both grammatical and ungrammatical "English" sentences.

P(E) The Bayesian condition shown above requires also that the generated translation be assigned a probability of language correctness. This implies the development of a language model for English. Typical language models in current use for this sort of application involve bigram or trigram distributions.

As a check on the quality of output resulting from using the training material, a held out portion of the corpus, perhaps 5 or 10 per cent, is decoded according to the training data organized according to Equation 1. The resulting translations are assessed against the corpus pairs, perhaps using a word error rate metric based on Dynamic Programming using Levenstein Distance (Levenstein 1966), or perhaps a more complex metric based on multigrams such as that of Papineni et al (2001).

One thing which became clear about this method of Machine Translation is that the quality and size of the input corpus is all important, and a vigorous literature has developed concerning how to align a corpus into sentence pairs with a high degree of word correspondences. Alignment methods proposed include sentence length (Gale and Church 1991), dictionary correspondence (Kay and Roscheisen 1993), and hybrid methods (Haruno and Yamazaki 1996). Indeed this lack of suitable parallel corpora seems to be one of the limiting factors in the spread of SMT. A second limiting factor is the computational intensity of

the decoding phase. Typically sentences of lengths up to around 10 words take only a few seconds to decode, but 12 and 13 words take up to an hour, and 14 plus word sentences are impractical for most MT users. These factors are leading to more careful consideration of Example Based Machine translation, introduced next.

1.2 Example Based Machine Translation

Inspired by Nagao (1984), EBMT adopts an analogical, or case based approach to translation using a bilingual corpus. Unlike SMT, there is no strong mathematical theory behind the development and use of EBMT systems (it is therefore relatively quick and simple to develop an EBMT system), and there are various formulations of how it should be done. The approaches of [Sumita and Iida 1991], [Cicekli and Guvenir 1996], [Veale, T and Way, A 1997] and [McTait 2000] are representative of the progress in this field. Essential phases however seem to involve "Compilation" - analogous to Training in SMT, and "Recombination" - analogous to decoding in SMT.

Compilation involves identifying variables and templates in the source corpus, and cross-referencing them. Variables need not be Word or Phrase types, but strings discovered by string-matching. Templates might be sentence skeletons with verbs, adverbs and function words, and references to NPs, PPs, linked to their correspondents in the other language. These templates may also be broken down into segments, each correlated with its translation.

Recombination entails decomposing a Source sentence into template(s) and variables, and matching them against the Example base, combining and filtering to find the best candidates.

The big advantage that EBMT has over SMT is that its results are more naturalistic, because the compilation phase retains the grammatical integrity of the sentence and its translation, even in template form, and a balanced corpus requires a broad range of structures, while high frequency is less important. However it suffers the same disadvantage in that a sparse corpus compiles into an impoverished structure with word and phrase translations missing. Therefore automatic translation leads to at best omission of variables, at worst random substitution of inappropriate variables. For these reasons, a supporting translation method of Machine Aided Human Translation maybe used. This is Translation Memory, discussed next.

1.3 Translation Memory

There is a broad overlap between EBMT and Translation Memory in that both require an aligned bilingual corpus, and have Compilation and Recombination phases. The main difference is that while EBMT is aimed at FA(HQ)MT, TM is pitched at Machine Aided

Human Translation (MAHT) [Kay, M 1981]. Consequently the corpus isn't initially expected to be as comprehensive as an EBMT corpus.

The idea behind TM is that the results of Recombination are presented to the user as a selection task, and where translations are unavailable, the user has the option of supplying them. TM is especially useful when the corpus is small, and when there is a high incidence of unknown words and variables. It is a prudent step to implement a Translation Memory system as a stage in the development of a fully automated EBMT system.

1.4 Structure of the Report

This report is about experiments in corpus processing for Machine Translation.The corpora are introduced in Section 2. The tools used to conduct the experiments include publicly available programs and locally developed programs. The public tools are documented in Section 3 and the local tools in Section 4. The experiments are introduced and described in Section 5. Some concluding remarks are given in 6, and the Appendix includes an inventory of corpora, programs, configuration files and results.

2 Corpora

The corpora which have been used in the course of this investigation include the ATR Basic Travel Expressions Corpus(BTEC), the NHK News articles (1995 - 2000) and Nikkei News articles (July 2000). These are described here in Subsections 2.1, 2.2 and 2.3 respectively.

2.1 ATR Basic Travel Expressions Corpus

The ATR BTEC corpus includes English and Japanese alternants of sentences and expressions used in foreign travel situations. These are typically short question and answer type expressions with an average length of about 6 words. There are 139,478 unique sentence pairs in all. Some examples are given below.

1(a) 今 チェックイン できますか。

1(b) can i check in now .

2(a) 部屋 を 替え て ほしい の ですが。

2(b) i 'd like to change rooms .

3(a) 至急 だれ か 来 て下さい。

3(b) send someone quickly to my room .

4(a) 部屋 に 鍵 を 忘れ ました。

4(b) i locked myself out .

5(a) 鍵 が 壊れ ています。

5(b) the key to my room doesn't work .

In these kinds of expressions the translations are rather direct, so they should provide a good basis for Statistical Machine Translation. Still, on inspection the word for word correspondence is not so direct. The literal gloss for the Japanese of 1(a) is "now check in possible (question)", which is more directly translated as "Is it possible to check in now", so there is need to draw a correspondence between "can I" and "できますか". Item 4, "room in key forgot" is a straightforward, but not a literal translation of "I locked myself out", and there is a need to make the correspondence between the verbs 忘れました (forgot) and "locked ... out".

This is a rather straightforward corpus in a format suitable for direct input to the Statistical Machine Translation process.

2.2 NHK News

The NHK News Corpus [Tanaka et al 2002] includes 41.1K content-aligned news articles of Japanese and their free English translations. Each article has typically 4 - 10 sentences, and while the average length of a Japanese sentence is about 40 words, the English sentences average about 21 words in length. A typical example article pair concerns the debate in Japanese about the introduction of Daylight Savings Time, illustrated in 1 below. The first sentence of each article is given below, in which the English has 19 words and the Japanese about 36. Both sentences concern a meeting in Tokyo to discuss the introduction of daylight savings time to Japan, but the Japanese 有識者 (yuu shiki sya = "experts") expands to "representatives of labor business and consumer groups". Moreover the Japanese sentence contains an extra clause concerning the possible introduction of legislation, which accounts for part of the additional length. The notion of Daylight Savings Time is not well known in Japan, but widely used elsewhere, so the Japanese explanation of what it is, is omitted from the English. In the translation of this sentence then, there are some corresponding terms and some redundant elements. To a greater or lesser extent, the same situation holds true throughout the corpus, so reducing its utility for high quality statistical machine translation.

1(a) Representatives of labor business and consumer groups have met in tokyo to discuss introducing daylight saving time to japan

1(b) 夏の間 だけ 時計の針 を 一時間 進める サマータイム 制度 を
summertime only clock OBJ one hour advance summertime system OBJ

日本 で も 導入する よう 求める 有識者らの 会合 が
Japan to also introduce like ask for experts meeting NOM

きょう 東京 で 開かれており 国会 が 立法化 に 向けて
today Tokyo in was opened Diet NOM legislation DAT facing

審議 に 入る よう 訴える アピール など を 採択します
discussion DAT enter like appeal appeal other OBJ adopt

2.3 Nikkei News

The Nikkei News Corpus [Tanaka et al 2002] is similar in structure to the NHK News. In all there are 5 years of articles with 1.8 million Japanese articles containing 18.6 million sentences, and 184K English articles containing 1.4 million sentences. Average sentence lengths in both cases are 21 words. Of these articles only one month's supply has been content-aligned, so the parallel corpus contains 1929 articles from July 2000.

2.4 Dictionaries

For confirming word senses between Japanese words and their English translations, we have been using a dictionary which is the combination of three on-line dictionaries. The component dictionaries are Edict [?] containing 97.7K entries in the form given here:

Edict : 伺い [うかがい] /(n) inquiry/question/call/consulting the oracle/visit/

Enamdict containing 201K names, including personal names and place names in a similar form:

Enamdict : 興道 [こうどう] /Koudou (g)/

and Eijiro, a large dictionary of 1 million entries, including single word lookups, and also expressions such as the following:

Eijiro ひょうが降る =It hails

These are combined together in a canonical form in a "mega-dictionary" with 1.2 million entries. The form is:

Megadict むき出しになった根元 =bared_root

3 Tools: Public

The experiments described in Section 6 of this report rely on the use of corpus processing tools, both publicly available and locally built. This section introduces the public tools used in the experiments. These include taggers and parsers for both Japanese and English, an English language modelling kit, and a bilngual SMT training kit.

3.1 Brill English Tagger

The Brill Tagger was developed for tagging English text with Part of Speech (POS) tags, using the Penn Tags and trained on Brown and Wall Street Journal corpora. It is a rule based tagger for which overall claimed accuracy is 94%. An example of text before and after tagging with Brill is given in 1(a) before and (b) after.

1. The English is puzzling .

2. The/DT English/NNP is/VBZ puzzling/JJ ./.

The overall accuracy claimed for the Brill tagger is about 94%, but since it was trained on Brown/Wall Street corpora, accuracy can vary when used with other corpora. An analysis shows that noun accuracy is about 98%, while accuracy for verbs can be less than 80%. Brill can be retrained on local corpora, and new rules written to accommodate newly identified tagging criteria. Where a new corpus includes named entities and foreign words (外来語) substantially different from the original corpora, Brill should be retrained to get optimal relevant results.

The Brill Tagger is available from: http://www.cs.jhu.edu/ brill/

3.2 ChaSen Japanese Tagger

ChaSen takes any Japanese text as input, splits it into morphemes based on its associated dictionary. Morphemes are analysed one per line with their base form, actual form, reading (katakana), and complex part of speech. If ChaSen's dictionary does not include words found in the source text, ChaSen overanalyzes into sub-words, and individual Kanjis. For the best results it is therefore useful to add to ChaSen's dictionary words extracted from a source text.

ChaSen is available from: http://cl-aist-nara.ac.jp/lab/nlt/chasen.html

3.3 CMU Language Model

Proper decoding in Statistical Machine Translation is a function of Bayes law concerning the probability of translation of the Target language. In Japanese-English terms, P(T) is determined using an English language model, and the CMU toolkit
[Clarkson, P and Rosenfeld, R 1997] is available for this purpose. It is trained on the English half of the bilingual corpus used in translation model training (See Egypt, next). Bigram or Trigram language models can be trained. In principle a trigram language model leads to more accurate results, but it needs a much larger corpus. In the absence of a large enough corpus, a bigram language model produces adequate results.

The process usages and their sequences are well described in the CMU-LM home page at: http://swr-www.eng.cam.ac.uk/ prc14/toolkit.doc. The CMU Language Model Toolkit is also available from there.

3.4 EGYPT

The IBM Statistical Model proposed by Brown et al is a practical implementation of the Bayes Rule formulation shown above. It is further decomposed into 5 models, one feeding into the next. The 5 Models address "The probability of the Target given the Source" - P(T|S) based on word counts, based on word alignments within paired sentences, and based on word fertility models. The Egypt package (Al Onaizan et al 2000), extended by Giza++ (Och 2001) is an implementation of the IBM SMT system. Its principle components are: Whittle, Giza/Giza++ and Mkcls. Egypt is available from: http://www.clsp.jhu.edu/ws99/projects/mt/toolkit/. The package includes instructions on how to prepare data and run the system. The Giza++ extensions can be downloaded from: http://www-if.Informatik.RWTH.Aachen.de/web/Software/GIZA++.html.

3.5 Charniak Parser

A traditional way to analyse a language is with a sentence-by-sentence phrase structure parse. While strict rule-based systems are not fully able to cope with the full potential range of structures, a statistically trained corpus based parser develops sentence-by-sentence parses with rules conditioned on the training. Phrasal structures can be extracted from these structures and used as input to other packages, such as Egypt. The Charniak parser offers the highest precision/recall of any freely available parser, at 90.1% (see [Charniak 2001]) Depending on length, a Charniak parse takes 3 - 5 seconds to run per sentence. A corpus of 150K sentences will then take about 168 hours, or a week to run.

The Charniak parser is available from: ftp://ftp.cs.brown.edu/pub/ nlparser/

3.6 CaboCha Parser

As an alternative to Phrase Structures, a dependency analysis is also a viable way to analyse sentence structure, and is a popular way to analyses Japanese sentence structure in particular. CaboCha is a parses implemented using Support Vector Machines [Kudoh and Matsumoto 2000] which takes a corpus of Japanese sentences, and outputs a dependency analysis of each sentence. The form of output presentation preferred in the work described in this report is as an XML document. From this target data, structures can be extracted which are roughly parallel to the structures from Charniak.

CaboCha is available from: http://cl-aist.nara.ac.jp/ taku-bu/software/cabocha/

3.7 Network Kanji and Conversion Filter (nkf)

While NKF is not particular to any of the experiments described in this document, it is a staple of Japanese language processing as it identifies the Kanji encoding of a file, and converts to any code specified by the user. Thus, it converts freely between JIS, Shift-JIS, EUC and Unicode, with a low (but non-zero) error-rate. Any experimental sequence of software processes should be aligned with respect to a single known code, and this is ensured by using NKF on the source corpus.

NKF is a system command on Japanese versions of Unix.

4 Tools: Local

4.1 cabfluff.pl

Command Line cabfluff.pl -c cabfluff.nhkeip

Description Remove the phrasal heads and POS information and print out streams of phrases, space-separated, with the phrases contained by each article identifier (MI=##199503060063) on a separate line. Accumulate frequencies of each unique phrase and print them out in the *.nccs_f file.

Input Lexicalized Japanese phrases. Examples:

MI=##199503060063

NP= 夏

NP= 間

NP= 時計

NP= 針

NP= 一 時間

VP= 進める

Output Bags of Phrases and List of Unique Phrases. Examples:

Bag:

夏 間 時計 針 一 時間 サマータイム 制度 日本 よう 有識者 ら 会合 きょう 東京 国会 立法 化 審議 よう アピール これ 労働 界 産業 界 それ 消費 者 団体 代表 ら 日本 ゆとり サマータイム 会 もの です 会 会長 元 経済企画庁 長官 高原 須美子 さん ゆとり 暮らし ため サマータイム 導入 サマータイム 制度 の 昼間 時間 夏 間 時計 針 一 時間 もの です 勤務 時間 帯 実質 的 一 時間 こと 朝 時間 有効 冷房 電力 節約 余暇 利点 世界 七十 ヵ国 以上 国 会合 今 国会 中 サマータイム 制度 立法 化 審議 よう アピール 予定 です が 動き 国会 立法 化 ため 超党派 議員 連盟 動き 今後 サマータイム 制度 論議

Unique:

橋本 総理 大臣 5172

協議 4924

韓国 4273

きのう 4219

方針 4154

意見 4030

現在 3785

来月 3758

小渕 総理 大臣 3617

Config File cabfluff.nhkeip

newsdir=/home/pxs103/sltuser/night/corpora/nhk/mwvcs/ ## Input directory.

jparsed=eip.nplex_f ## Input file of lexicalized phrases.

jgranular=eip.nps_f ## Output file of bags of phrases.

jncc=eip.nccs_f ## Output file of Phrases and their frequencies.

keeplist=NP—MI ## Which phrases to keep.

progname=cabfluff ## Program name.

4.2 decoder

For details see Taro Watanabe.

4.3 decoderesult.pl

Command Line decoderesult.pl -c dtest_e -x plotnonuwer ⟨ decoder-log ⟩ nonuwer.html

Description Given the decoder-log and the English sources from the tests set, generate a table containing the Japanese and English sentence pair, the decoder output (English translation), time to decode, sentence length and the Word Error Rate.

Input The Decoder log file generated by Watanabe-san's SMT Decoder:

⟨?xml version="1.0" encoding="euc-jp" ?⟩

⟨decoder algorithm="beam"⟩

⟨decoder-result⟩

⟨time seconds="148.359" /⟩

⟨channel-target⟩ あなた の フライト を 確認 いたし まし た 。 ⟨/channel-target⟩

⟨decoder-result-item nbest="1" score="1.3614e-24" ⟩

⟨channel-source⟩would you had confirm my flight . ⟨/channel-source⟩

⟨alignment⟩2 0 6 0 4 1 3 3 7 ⟨/alignment⟩

Output Decoded and Scored HTML. See tables on pages 17-18.

Command Line Arguments -c: English Sentences from the test set.

-x: Output plot parameters.

4.4 defluff.pl

Command Line defluff.pl -c defluff.eip

Description Remove all superfluous phrasal types and print out streams of lexicalized words. Save the unique phrases and their frequencies.

Input Lexicalized English Phrase Structure rules:

QP=## 199503060007 ;

NP=QP ;

NP=representatives ;

NP=labor business and consumer groups ;

NP=NP PP ;

NP=tokyo ;

NP=daylight saving time ;

NP=japan ;

NP=her opening address ;

NP=monday ;

Output Bags of Phrases and List of Unique Phrases. Examples:

Bag:

representatives labor business and consumer groups tokyo daylight saving time japan her opening address monday former economic planning agency director general takahara meeting introduction daylight time best use daylight hours life daylight saving clock hour summer daylight hours work time hour earlier cooler morning hours energy air conditioners daylight saving better opportunities recreation darkness system more than seventy countries worldwide group appeal government deliberations daylight time current parliamentary session members parliament non partisan group system debate

Unique:

nikkei average 1450

bill 1418

market sources 1417

she 1409

okinawa 1389

iraq 1381

225 selected issues 1379

tokyo foreign exchange market 1368

mr obuchi 1343

Config File newsdir=/home/pxs103/sltuser/night/corpora/nhk/mwvcs/ ## Input directory.

eparsed=eip.nplex_e ## Input file of lexicalized phrases.

egranular=eip.nps_e ## Output file of bags of phrases.

encc=eip.nccs_e ## Output file of Phrases and their frequencies.

progname=defluff ## Program name.

4.5 eonly.pl

Command Line eonly.pl ⟨ lexicalized.snt ⟩ lexicalized_e

Description Given the training or test file generated by whittle.perl, create a file with English sentences only, with bigram or trigram context cues.

Input The lexicalized output from whittle.perl

1

please speak slowly .

ゆっくり 言っ て下さい 。

2

where is the boarding gate .

搭乗 ゲート は どこ です か 。

Output The English sentences with context cues:

⟨s⟩ ⟨s⟩ please speak slowly .⟨/s⟩ ⟨/s⟩

⟨s⟩ ⟨s⟩ where is the boarding gate .⟨/s⟩ ⟨/s⟩

Config File None. No arguments.

4.6 flatcharn.pl

Command Line flatcharn.pl ⟨ charnout_e ⟩ flatcharnout_e

Description Convert the output of Charniak's parser to one per line.

Input Parsed sentences:

(S1 (S (NP (NNP daylight) (NN saving))

(ADVP (RB also))

(VP (VBZ gives)

(NP (NP (JJR better) (NNS opportunities))

(PP (IN for) (NP (NN recreation))))

(SBAR (IN as)

(S (NP (NN darkness)) (VP (VBZ falls) (ADVP (RBR later))))))
(. .)))

Output Sentences flattened to one per line:
(S1 (S (NP (NNP daylight) (NN saving)) (ADVP (RB also)) (VP (VBZ gives) (NP (NP (JJR better) (NNS opportunities)) (PP (IN for) (NP (NN recreation)))) (SBAR (IN as) (S (NP (NN darkness)) (VP (VBZ falls) (ADVP (RBR later)))))) (. .)))

Config File None. No arguments.

4.7 makegenre.pl

Command Line makegenre.pl -c makegenre.economy

Description Read articles of the given genre from the English (e.txt) and Japanese (j.txt.euc) corpus files and write them to genre specific files: e.txt.*genre* and j.txt.*genre*.

Input English and Japanese NHK source articles.

Output E and J genre files.

Config File progname=makegenre
thegenre=economy
genrecontrol=j.ref
newsdir=/home/pxs103/sltuser/night/corpora/nhk/
runpath=/home/pxs103/sltuser/night/
jarticles=j.txt.euc
earticles=e.txt
joffsets=newjxref
eoffsets=exref

4.8 newlexicalize.pl

Command Line newlexicalize.pl -c newlexicalize.nhkeip

Description Given parsed sentences, extract all phrase structure rules, and lexicalize the tagged elements as controlled by the go list in the Config file. Print out only Phrasal types as defined by the keeplist.

Input

Output

Config File termtags=$PRP\$|PRP|CD|EX|PDT|DT|FW|NN|NNS|NNP|NNPS|$
$POS|JJ|JJR|JJS|RB|RBR|RBS|RP|VB|VBD|VBG|VBN|VBP|VBZ|SYM|$
$UH|IN|MD|AUX|AUXG|CC|TO|WP|WP\$|WRB|WDT|.|$:
nontermtags=.|$ADJP|ADVP|CONJP|FRAG|INTJ|LST|NAC|NP|NX|$
$PP|PRN|PRT|QP|RRC|S|S1|SBAR|SBARQ|SINV|SQ|UCP|VP|WHADJP|$
$WHADVP|WHNP|WHPP|X|G1|G2|G3$
newsdir=/home/pxs103/sltuser/night/corpora/nhk/neweip/
eparsed=nhkeip.flatcharn_e
egranular=nhkeip.lex_e
golist=$UH|CD|EX|FW|NN|NNS|NNP|NNPS|POS|JJ|JJR|JJS|VBG|VBN$
keeplist= $NP|QP$
progname=newlexicalize

4.9 newwhittle.pl

Command Line newwhittle.pl -c newwhittle.eip

Input (1)English and Foreign 'bags' of parallel texts.
(2) English and Foreign phrases.

Output (1) Tokenized 'bags' file: Example:
(2) Tokenized English Vocabulary: Example:
(3) Tokenized Foreign Vocabulary: Example:

Config File progname=newwhittle
corpusdir=/home/pxs103/sltuser/night/corpora/nhk/mwvcs/
ecorpus=eip.nps_e
fcorpus=eip.nps_f
trainpart=100
testpart=0
fullcorpus=1
bigdict=/home/pxs103/sltuser/night/corpora/dictionaries/je.alledict
encc=eip.nccs_e
fncc=eip.nccs_f
nccdict=

Description Investigations of the EGYPT tools [Al Onaizan et al 2000, Och and Ney 2000] for SMT show that the surface details of the two languages are abstracted away from the problem of identifying translation candidates (or token alignments). The Whittle corpus pre-

English Compound	Frequency
civilians	110
chemical weapons	110
foreign ministry spokesman	109
Japanese Compound	**Frequency**
合意 文書	205
ポル・ポト 派	204
青木 官房 長官	204

表 1: Noun Phrases

processor tokenizes the vocabulary in source and target corpora and maps words to tokens 1-to-1. An example of input is given in (1), with a sentence from the Japanese input file in 1(a) and its translation from the English input file in 1(b). Whittle tokenizes the vocabulary and maps words to tokens as in (2) (a) and (b), where each unique word form associates with a single value.

1(a) 冷やし た ミネラルウォーター の 小 ビン を 持っ てき てください 。

1(b) bring me one small bottle of chilled mineral water please .

2(a) 5899 14 1303 6 1383 4031 8 97 143 16 2

2(b) 195 18 39 231 547 24 3857 1101 169 9 2

Giza, the EM engine takes these token vector pairs and performs the IBM Model 1-5 transformations [Brown, et al 1993] to generate a translation model. Because of the abstraction, Giza only knows about these token vectors and not about words and sentences. It is perfectly feasible therefore to map more complex word groups onto single tokens, in particular parsed phrasal structures. Although we ultimately departed from EGYPT, this was one of the inspirations for our modularization discussed below. Because we extracted phrasal chunks from the corpus, these are tokenized with many words mapped to one token (4.1). Candidate generation proceeds with these token vector pairs (4.2), followed by dictionary filtering (4.3).

Flexible Tokenization

Newwhittle takes as input pairs of word strings, which may be phrases or sentences, articles or paragraphs: the surface form can be identical to that taken by Whittle. These need not be complete sentences, the primary sources are filtered for salient features such as Noun Phrases, discarding verbal phenomena and function words. Newwhittle also takes as

input lists of word groups, such as NPs, to be identified in the source strings, and mapped onto single tokens. Examples (not translation pairs) of such NP lists are given in Table 2, with determiners and postpositions pruned. Nouns occurring singly (e.g *civilians*) are not excluded.

The strategy is to first tokenize individual words, and separately tokenize the word group lists, then map words to tokens 1-for-1, as per classic Whittle to create token vectors. In a second pass, the token sequences from the word group lists are substituted into the token vectors, thus compressing them. As example, the sentence pair in 1(a) and (b) can be filtered for Noun Phrases, yielding the result in 3 (a) and (b), and mapped onto tokens in 4(a) and (b). The tokenized word groups are then remapped to give the tokenized NP sequences in 5(a) and (b), where for example the string "one small bottle" maps to the unique token 9991.

3(a) (冷やし た ミネラルウォーター) (小 ビン)

3(b) (me) (one small bottle) (chilled mineral water)

4(a) (5899 14 1303) (1383 4031)

4(b) (18) (39 231 547) (3857 1101 169)

5(a) 9990 9999

5(b) 18 9991 9992

Following translation candidate generation we would expect to see the pairings:

小_ビン = *one_small_bottle*, and

冷やし_た_ミネラルウォーター =

chilled_mineral_water

identified as translation pairs.

4.10 normalize.pl

Command Line normalize.pl ⟨ corpus_e ⟩ corpus_norm_e

Description Read English text articles from the input file and perform various text normalizations, such as lower casein everything, removing extraneous punctuation.

Input An English text file.

Output A normalized English text file.

4.11 retro.pl

Command Line retro.pl -c retro.eip

Description Read the file of translation candidates generated by Giza (or by any program which generates candidates in the format: 'Japanese English Number'), and filter all translation pairs using the dictionary. There are three modes of granularity and these are: Relevant, Unique and Absolute. If mode is 'Relevant', then all candidates are accepted which have at least a one word dictionary lookup match. If mode is 'Absolute', then only candidates with an exactly matching number of words, which match entries in the dictionary, are accepted.

Input A file containing English and Japanese translation candidates.

Output A file containing English and Japanese translations.

Config File progname=retro ##The name of the Perl script.
corpusdir=mwvcs/ ##The directory containing the data.
probabilities=102-12-17.111717.night.actual.ti.final ##The input translation candidates.
bigdict=dictionaries/je.megadict ##The E/J dictionary to filter translations.
outdict=eip.npdict ##The output translations file. mode=Unique

4.12 splitsort.pl

Command Line splitsort.pl ⟨ test.snt

Description Splitsort.pl reads the Whittle test output file, sorts the sentences by increasing length and prints to separate English and Foreign sentence files in batches of 100.

Input Whittle test corpus output format.

Output English and Foreign sentences in separate files.

4.13 xcab.pl

Command Line xcab.pl ⟨ cabochaxml_f ⟩ phrasetypes_f

Description Xcab.pl reads the XML file generated by CaboCha containing sentences chunks and tokens, puts the chunks back together, infers a head for each chunk, which may be Noun, Verb, Adjective or Other, and prints out the lexicalized phrases.

Input XML output from CaboCha.

Output Lexicalized phrases chunked together:
MI=##199503060063
NP=夏
NP=間
NP=時計
NP=針

NP= 一 時間

VP= 進める

NP= サマータイム 制度

NP= 日本

VP= 導入 する

5 Experiments and Results

The sequence of experiments described in this section developed following the experience of finding that available corpora are not directly relevant to Statistical Machine Translation. For the first experiment, setting up the process sequence for Statistical Machine Translation, this was established using the ATR BTEC corpus, a corpus of travel phrases with a high degree of literal translation. This corpus has been in use at ATR for some years and the objective of Department 4 is to develop Machine Translation methods for News, and therefore News Corpora are more applicable.

Following the discussion in Section 2, it is notable that News articles contain longer sentences than travel phrases, and are more 'free' translations, with a lower degree of literal overlap. Together with the fact that news sources are in article pairs, without a 1-to-1 sentence correspondence it becomes clear that some corpus pre-processing is necesssary to get sentence-aligned. The objective of experiment two is to find aligned phrases in the NHK corpus, suitable for use in a bilingual phrase dictionary.

The overall level of phrasal translational equivalence in the NHK corpus is disappointingly low, and extending the alignment method to find Japanese and English translationally equivalent sentences yields a very small corpus indeed. However the level of literal translation in the Nikkei News Article corpus is higher, a greater proportion of phrasal translations can be extracted, and a useful yield of alignable sentence pairs is seen. The third experiment in this section develops the alignment method for these sentence pairs.

5.1 Setting Up Statistical Machine Translation

The objective of this experiment is to develop a process sequence for Statistical Machine Translation using the ATR BTEC corpus (See Section 2.3) and make a baseline evaluation. This entails installing and deploying public tools, developing local corpus processing and results processing tools, and evaluating the results. SMT processing phases include: (1) Corpus Processing; (2) Language Model Training; (3) SMT training; (4) Decoding and Results Analysis. The detailed process sequence including configuration information is described in 5.1.1 and its results and consequences discussed in 5.1.2.

(5.1.1) Process Sequence

Corpus ATR Basic Travel Expressions Corpus, 146K English-Japanese sentence pairs, split into 90% training and 10% test.

Public Tools :

whittle.perl, mkcls, giza++, text2wfreq, wfreq2vocab, tex2idngram, idngram2lm, evallm, gnuplot.

Local Tools :

nonu.pl, eonly.pl, splitsort.pl, decoder, decoderesult.pl.

Corpus Processing :

nonu.pl ⟨ phrasebook_e ⟩ phrasebook_nonu_e

whittle.perl -w -b 0.1 -t 0.9 ⟨ Nonu/

whittle.perl -b 0.1 -t 0.9 ⟨ Nonu/

splitsort.pl ⟨ phrasebook_nonu-test.snt

eonly.pl ⟨ phrasebook_nonu-train.snt ⟩ phrasebook_nonu_train_e

eonly.pl ⟨ phrasebook_nonu-test.snt ⟩ phrasebook_nonu_test_e

Language Model Training :

text2wfreq ⟨ phrasebook_nonu_train_e ⟩ ephrasestrain.wfreq

wfreq2vocab ⟨ ephrasestrain.wfreq ⟩ ephrasestrain.vocab

text2idngram -n 3 -vocab ephrasestrain.vocab ⟨ phrasebook_nonu_train_e ⟩ ephrasestrain.id3gram

idngram2lm -idngram ephrasestrain.id3gram -vocab ephrasestrain.vocab -n 3 -binary ephrases-train.3gram.binlm -cutoffs 1 1 -witten_bell -context ephrasestrain.ccs

evallm -binary ephrasestrain.3gram.binlm

SMT Training :

mkcls -c80 -n1 -pphrasebook_nonu_e -Vphrasebook.e.vocab.classes

mkcls -c80 -n1 -pphrasebook_f -Vphrasebook.f.vocab.classes

geezer geezer.config

Decode and Results Analysis :

decoder –config decoder_nonu.config –print-xml –time true ⟨ dtest10_f — tee decoder-log10

decoderesult.pl -c dtest10_e -x plotnonuwer10 ⟨ decoder-log10 ⟩ nonu.wer10.html

gnuplot, output to plotnonuwer10.ps

(5.1.2) Discussion and Results

The ATR BTEC corpus requires a few small modifications to make it completely SMT ready: these involve removing underscores from collocated items, so we can get a clean baseline result, and after splitting the corpus into 90% training and 10% test, sorting the test set by increasing length, so we can graduate the load on the Decoder. The Decoder combines trained corpus information from the Language Model (from CMU LM) and the MT model (from Giza++). The corpus inputs to these models differ slightly, so in corpus

processing Whittle is run twice: once with a '-w' switch to generate lexical output for the CMU LM, and once without to generate tokenized output for Giza++. The output from 'whittle.perl -w' is run through 'eonly.pl', to generate an English only corpus with context cue tags set, suitable for a trigram language model. The LM training model sequence generates a trigram language model for English, with a Perplexity value of 11.66.

TM training includes running 'mkcls' and running 'giza'. Mkcls is run for both English and Japanese vocabularies to establish word classes which help in the efficient processing of the translation model by Giza++. Experimentation with various numbers of classes shows that 120 is the optimum number of classes, as measured by the final Word Error Rate in Results Analysis (See Below). Input parameters to Giza++ are drawn from the giza.config file. These include setting the number of iterations of Models 1 to 4 (20 iterations for Models 1,3 and 4 and 0 iterations for Model 2), selecting the input corpus file phrasebook-train.snt, E and J vocabularies phrasebook.e.vocab and phrasebook.f.vocab, and their vocabulary classes. After training the models, Giza++ outputs files for: translation probabilities, distortion probabilities, fertilities, perplexities for each iteration, Source and Target vocabulary files for both training and test corpora, a file containing initial configuration values, and a Decoder configuration file. The contents of these files are described in the Giza++ documentation.

Decoding and Results Analysis requires configuring and running the ATR decoder (Watanabe), and analysing results with decoderesult.pl. Results for some of the sentences, all of length 9 words, are given in Tables 1 and 2. Each entry has the Japanese target sentence and matching English Source followed by the Decoder output. Word Error Rates are calculated based on the difference between English source and Decoder output. Decoding time in seconds is given for each sentence. In these samples the Word Error Rates vary between 0.143 (respectably low) and 1.6 (unacceptably high). The overall average for all sentences decoded is 0.58. Example 1 has the highest WER, though the sense of the translation is okay. The decode result is grammatically incorrect though. The best example, number 17 with a WER of 0.143 has just one word mistranslated, where 'is' replaces 'does'. The result can be easily understood, but again is ungrammatical. Examples containing translated numbers fare less well. In 10 and 11 the correct translation for 九十九 is ninety nine, but the translations are 'nine' and 'nine thirty' respectively. Although the word errors may be small here, the mistranslation is significant, more so than non-numeric word translations. It makes little sense to use an MT system for number translation, when a rule based system or digital representation in both languages solves the problem. For this reason I strongly recommend preprocessing the training and test corpora to normalize all numbers.

Overall, the effect of translation gives a reasonable sense of the correct meaning, but there

are enough unfortunate mistranslations that it cannot be relied on as an automatic tool. Decoding time also poses a problem. For these sentences, all of length 9, the variation is between 36 and 135 seconds. An average of 90 seconds is an uncomfortably long time to wait for a result, and for longer sentences this turns into hours and days.

The preliminary conclusion from running an easy corpus such as the ATR BTEC over the Statistical Machine Translation system is that the method needs some improvements in grammatical transfer, and in performance.

Sq	Japanese / English Source / Decoder	WER	Lng	Decode
1	あなた の フライト を 確認 いたし ました 。 your flight is reconfirmed . would you like to confirm my flight was it .	1.600	9	119.
2	現金 と カード どちら に なさい ます か 。 cash or charge . cash or credit .	0.250	9	45.1
3	少々 お 待ち ください 。 調べ てまいり ます 。 just a moment please . i 'll check . check . hold on the back .	0.889	9	127.
4	新製品 は 発売 さ れ てい ます か 。 are there any new items on sale . do you have items .	0.750	9	50.6
5	香り が 強く ない もの を 選ん でください 。 please pick something which is not so strong . not so strong wine .	0.667	9	130.
6	食後 コーヒー か 紅茶 は いかが です か 。 would you like coffee or tea after the meal . coffee or tea or after .	0.600	9	37.1
7	この 切符 を 普通車 の に 代え たい 。 i want to change this ticket to a coach ticket . i 'd like to change this ticket .	0.545	9	135.
8	サラダ と デザート が つく ステーキ セット です 。 it 's hillman's steak which comes with salad and dessert . set guide for a dessert steak and a salad .	0.909	9	83.5
9	エービーシーホテル 行き の バス は あり ます か 。 is there a bus to the abc hotel . i 'd like to have a bus go to the abc hotel .	0.667	9	53.8
10	どれ で も 九 十 九 ドル です 。 they are all ninety nine dollars . how long nine dollars .	0.571	9	36.1

表 2: SMT Results (1 - 10)

Sq	Japanese / English Source / Decoder	WER	Lng	Decode
11	三 個 で 九 十 九 ドル です 。 ninety nine dollars for three . nine thirty dollars each .	0.667	9	42.5
12	ヒル ホテル です か 。 すみません わかり ません 。 the hill hotel . sorry i don't know . excuse me . i can't dunhill hotel .	0.778	9	125.
13	窓側 の 席 を お 願い し ます 。 please make the seat by the window . window seat please .	0.750	9	80.5
14	郷土 の お みやげ 品 を 見せ てください 。 a local souvenir please . could you show me the local souvenirs .	1.400	9	135.
15	バス 付き ダブルルーム を お 願い し ます 。 i 'd like to have a double room with a bath . double with bath please .	0.750	9	50.
16	何 時 が 都合 よい の です か 。 when is a good day . i 'd like a good time .	0.667	9	55.2
17	この バス は ディズニーランド に 行き ます か 。 does this bus go to disneyland . is this bus go to disneyland .	0.143	9	40.0
18	窓 を 開け て も いい です か 。 may i open the window . may i open the window or charge .	0.333	9	42.1
19	ステーキ の 焼き方 は どのように いたし ます か 。 how would you like your steak . i 'd like steak .	0.571	9	44.3
20	どこ で 作ら れ た もの です か 。 where is this made . where was it made of .	0.600	9	67.8

表 3: SMT Results (11 - 20)

5.2 NHK Phrasal Extraction

The objective of this experiment is to find translationally equivalent phrases in the 41K aligned article pairs of the NHK corpus. Because the articles are content-aligned, not sentence-aligned, this involves parsing the articles sentence-by-sentence, extracting (in this case) Noun Phrases, and putting them in parallel bags, then using the EM algorithm to find translation candidates. The detailed process sequence including configuration information is described in 5.2.1, Discussion and Results in 5.2.2.

(5.2.1) Process Sequence

Corpus NHK News, Economy, International and Politics genres. 29.6K article pairs.

Dictionaries Edict (97927 entries) + Enamdict (210307 entries) + Eijiro (1031965 entries) = Total: je.megadict (1236854 entires).

Public Tools :

parseIt, CaboCha, mkcls, giza++

Local Tools :

flatcharn.pl, newlexicalize.pl, xcab.pl, defluff.pl, cabfluff.pl, newwhittle.pl, retrocheck.pl

Corpus Preprocessing :

makegenre.pl -c makegenre.economy

makegenre.pl -c makegenre.politics

makegenre.pl -c makegenre.international

normalize.pl ⟨ e.txt.economy ⟩ e.norm.economy

normalize.pl ⟨ e.txt.politics ⟩ e.norm.politics

normalize.pl ⟨ e.txt.international ⟩ e.norm.international

cat e.norm.economy e.norm.politics e.norm.international ⟩ nhkeip_e

cat j.txt.economy j.txt.politics j.txt.international ⟩ nhkeip_j

Corpus Processing :

parseIt DATA/ nhkeip_e ⟩ nhkeip.charn_e

cabocha -f3 ⟨ nhkeip_f ⟩ nhkeip.cab_f

flatcharn.pl ⟨ nhkeip.charn_e ⟩ nhkeip.flatcharn_e

newlexicalize.pl -c configs/newlexicalize.nhkeip

xcab.pl ⟨ nhkeip.cab_f

defluff.pl -c configs/defluff.nhkeip

cabfluff.pl -c configs/cabfluff.nhkeip

newwhittle.pl -c configs/newwhittle.nhkeip

Expectation Maximization :

mkcls -n2 -c120 -pnhkeip_e -Vnhkeip.e.vocab.classes

mkcls -n2 -c120 -pnhkeip_f -Vnhkeip.f.vocab.classes

geezer geezer.nhkeip (produce *actual.ti.final)

Results Extraction :

retrocheck.pl ⟨ 102-gizarun.actual.ti.final ⟩ nhkeip.filtered.dict

(5.2.2) Discussion and Results

Corpus Preprocessing: The original sources for the NHK articles are in file e.txt (73K English articles) and j.txt.euc (315K Japanese articles). Of these 41K pairs are article aligned, and the article/genre index is in file j.ref.genre. For each genre economy, politics and international, run makegenre.pl to extract article pairs by genre: 7543 for Economy, 9378 for Politics and 12701 for International. Normalize the English by lowercasing and deleting unnecessary punctuation. Combine these three genres into one pair of files, nhkeip_e containing 29.6K English sentences. Combine the Japanese genre files: j.txt.economy, j.txt.politics and j.txt.international into one file, nhkeip_j, containing 29.6K sentences. This is the starting point for processing the EIP subcorpus.

Corpus Processing: Parse the English corpus with Charniak and the Japanese with Cabocha and extract noun compounds. For English, NP extraction requires two processes: newlexicalize.pl and defluff.pl. The result is two files, nhkeip.nps_e with 29.6K 'bags' of NPs, and nhkeip.nccs_e, with 70389 unique Noun Phrases. For Japanese the processes are xcab.pl and cabfluff.pl, producing nhkeip.nps_f with 29.6K bags of NPs and nhkeip.nccs_f with 42549 unique NPs. These files are tokenized using newwhittle.pl, a rewrite of EGYPT's whittle.perl, which creates a tokenized corpus from the lexical inputs, with phrase to token mappings enabled by input of the unique NPs files. Newwhittle.pl gives the same outputs as whittle.perl: triples of tokenized English and Japanese sentences and the number of instances; and vocabulary files nhkeip.e.vocab with 25972 entries and nhkeip.nps.f.vocab with 32387 entries.

Expectation Maximization: Since this experiment is concerned with finding the optimal conditions for extracting parallel resources from noisy corpora, Giza++ is run multiple times to accommodate different training conditions: Run Model 1 only for 15 iterations; Run

Japanese	English
finance_minister_masayoshi_takemura	武村大蔵大臣
finance_minister_mitsuzuka	特別会計
finance_ministers	経済協力
finance_ministers_meeting	蔵相会談
finance_ministry_official	第二次中曽根改造内閣
finance_ministry_survey	調査結果
financial_assistance	資金援助
financial_authorities	行政当局
financial_authority	電力消費
financial_big_bang	金融ビッグバン
financial_burden	財政負担
firm_attitude	強硬姿勢
firm_opposition	対決姿勢
firm_recovery_track	回復軌道
firm_stance	強硬策
first_anniversary	一周年
first_appearance	初公判
first_asia	アジアヨーロッパ首脳会議
first_auction	第一回目
first_bank	ノンバンク GE キャピタル
first_batch	第一陣
first_business_day	今日午前
first_company	地域通信会社
first_conference	三者協議会
first_contingent	第一陣
first_country	欧米諸国

表 4: Translation Candidates

Models 1 and 2 for 15 iterations each; Run Giza++ modified to compute Mutual Information scores, for 1 iteration; and run Giza++ modified to compute exhaustively all translation candidates. The outputs are: 6419882 translation candidates for exhaustive; 1431849 for Model 1; 1155149 for Models 1 and 2; and 681084 for Mutual Information.

Results Extraction: The translation candidate generation methods given above all over-generate, and the numeric comparators cannot be relied on as a definitive method of gauging the 'correct' translation. For this reason a post-process dictionary lookup is implemented, with process retrocheck.pl. The number of Unique translation candidates discovered from the output of each Method is: 24504 for Exhaustive; 18619 for Model 1; 17690 for Models 1 and 2; and 18056 for Mutual Information. A full comparison and discussion of these results is given in the paper by Nightingale and Tanaka (2003). Some examples of the filtered Model 1 output are given in Table 4.

5.3 Nikkei Phrase and Sentence Extraction

The objective of this experiment is to find translationally equivalent phrases and aligned sentences in the 1929 aligned article pairs of the Nikkei corpus. The process sequence for Noun Phrase extraction is almost identical with that for extracting Noun Phrases from NHK. The process sequence for finding sentence alignments repeats some of the same elements with different arguments. The detailed process sequence including configuration information is described in 5.3.1, Discussion and Results in 5.3.2, with the discussion centering on the Sentence Alignment sequence.

(5.3.1) Process Sequence

Corpus Nikkei News 1929 article pairs.

Dictionaries Edict (97927 entries) + Enamdict (210307 entries) + Eijiro (1031965 entries) = Total: je.megadict (1236854 entires).

Public Tools :

parseIt, CaboCha, mkcls, giza++

Local Tools :

flatcharn.pl, newlexicalize.pl, xcab.pl, defluff.pl, cabfluff.pl, newwhittle.pl, retrocheck.pl

Corpus Preprocessing :

nikkei.pl ⟨ /data/D4L/user/kashioka/Nikkei-Align/Inter-Check/(nkm,nks,nss) ⟩ nikkei_e and nikkei_j

Corpus Processing :

parseIt DATA/ nikkei_e ⟩ charnout_e

cabocha -f3 ⟨ nikkei_f ⟩ cab_f

flatcharn.pl ⟨ charnout_e ⟩ flatcharnout_e

flatlexicalize.pl -c flatexicalize.nikkei.nps

xcab.pl ⟨ cab_f ⟩ nplex_f

defluff.pl -c defluff.nikkei.nps

cabfluff.pl -c cabfluff.nikkei.nps

newwhittle.pl -c newwhittle.nikkei

Expectation Maximization :

geezer geezer.nikkei.nps.ml

Results Extraction :

retro.pl -c retro.nikkei

Sentence Extraction :

lexheads.pl -c lexheads.nikkei

hitomorph ⟨ ssps_e ⟩ lemssps_e

hitomorph ⟨ ssccs_e ⟩ lemssccs_e

allxcab.pl ⟨ cab_f ⟩ heads_f

deptops.pl ⟩ sps_f, sscs_f newwhittle.pl -c newwhittle.nikkei3

geezer geexer.nikkei3

retrovp.pl -c retrovp.nikkei3

sortphrases.pl -t10 -b6

(5.3.2) Discussion and Results

The Nikkei aligned sentence corpus is created in two passes. The first pass comprises the extraction sequence for noun phrases. Initial corpus normalization is done with process nikkei.pl, which creates the clean article aligned corpora in the format of NHK. Subsequent processing includes parsing the English with Charniak and the Japanese with CaboCha, extracting the English NPs with flatlexicalize.pl and deffluff.pl, the Japanese NPs with xcab.pl and cabfluff.pl, and tokenizing the NP bags with newwhittle.pl. NP translation candidates are generated with Giza and filtered for literal translations using retro.pl. There are 25641 unique NP translation candidates. These are fed into the sentence alignment process.

For sentence alignment, we need to generate translation candidates which are sentence pairs, then check in the dictionary for any word translation overlaps. Verbs require dictionary form lookup, so these are lemmatized. The English sentences are glued back together from the output of flatcharnout, then lemmatized using the modified form of John Carroll's lemmatizer, called 'hitomorph'. The output from CaboCha is glued back together with allxcab.pl, retaining the lemmatized forms of verbs and adjectives. Unique sentences, and bags of sentences, are extracted with deptops.pl, and the combined Japanese and English corpora are tokenized with newwhittle.pl. Sentence translation candidates are generated using Giza++, with 2 iterations of the Expectation Maximization algorithm over word counts (Model 1). This output is filtered through the dictionary check by retrovp.pl and stored as Japanese and English sentence pairs with a count of the overlapping words. The function sortphrases.pl is run with upper and lower bound parameters to select sentence pairs with higher overlap counts, and therefore greater probability of alignment. There are 2825 sentence pairs with 7 or more overlapping words and NPs. Given 15K possible sentence pairs,

assuming 1-for-1 potential alignment, this offers a 13.5% recovery rate.

6 Summary, Conclusions and Recommendations

The goal of simultaneous interpretation from Japanese to English is extremely ambitious. Its component goal of Machine Translation is also hugely ambitious. Traditional, purely grammar based methods of MT did not scale up to the translation of full Natural Language. The introduction of Corpus-based methods brought in a new paradigm, and the past 12 years has seen a broadening of the effort to acquire corpora for new language pairs and to acquire much larger corpora, suitable for corpus-based MT paradigms. Shifting the paradigm from Grammar and Interlingua based methods to corpus-based methods has also shifted the proximate problem area. Statistical MT requires sentence-aligned corpora as input, and these must have highly consistent word and phrase pair alignments, or be very huge. The statistical training will promote the probability of consistently aligned word and phrase pair alignments, if they are present. There seems to be a relationship between the size of corpus needed and the amount of noise that SMT can subsume. This relationship is not yet quantified however.

The set of experiments and associated tools described in this report starts with the presumed availability of sufficiently aligned corpora, so that the results of the SMT process will be coherent. The results of training the ATR Basic Travel Expressions Corpus using the EGYPT machine translation tools shows a best average Word Error Rate of 0.58. For Consistently successful Machine Translation this figure needs to approach zero much more closely. The SLT Department 4 project is concerned with the translation of News, so training for Machine Translation with News Corpora is necessary. However the News corpora are article aligned, not sentence aligned. The subsequent experiments are aimed at achieving those sentence alignments. Results from the NHK corpus alignment experiments are very low, as a consequence of the fact that the literal word translation rate is low. It has been possible to extract some respectable noun phrase translations from the NHK News corpus however, thought the aligned sentences extracted are less than 1% of the corpus.

The results of Noun Phrase alignment, and Sentence alignment from the Nikkei corpus are much more promising. 2825 aligned sentence pairs represents 13.5% of the total alignable corpus. Since this is based on 1 month of articles, 1929 pairs in all, I strongly recommend that the Content-Alignment process for the remaining 5 years of Nikkei articles, be run. At the same rate of aligned sentence extraction it should be possible to get a corpus of some 150K aligned sentence pairs.

Because the SMT results from ATR BTEC were unsatisfactory, on a single word alignment basis, and because the sentence alignments from NHK and Nikkei were problematic, I suggest

that the results from these corpora be combined, and fed into continuing Example Based Machine Translation research. Fully Automated High Quality Machine Translation is an ambitious goal, and I also suggest that useful tangible results can be gained from introducing practical, near-term intermediate project goals. Developing a Translation Memory system on the way towards EBMT would be such a goal.

7 Appendix I: Location of Results files

All Perl sources and configuration files are in: /data/D4L/user/night/, directories bin and configs, respectively. Results from NHK and Nikkei corpus experiments are:

- **NHK Absolute NP Translations**
  /data/D4L/user/night/results/nhk.cnss.abs.np 805 pairs.
  /data/D4L/user/night/results/nhk.eip.abs.np 1359 pairs.

- **NHK Unique NP Translations**
  /data/D4L/user/night/results/nhk.cnss.unq.np 12254 pairs.
  /data/D4L/user/night/results/nhk.eip.unq.np 24504 pairs.

- **Nikkei Absolute NP Translations**
  /data/D4L/user/night/results/nikkei.abs.np 674 pairs.

- **Nikkei Unique NP Translations**
  /data/D4L/user/night/results/nikkei.unq.np 25641 pairs.

- **Nikkei Strongly Aligned Sentences**
  /data/D4L/user/night/results/nikkei.007.high.snt 2825 pairs.

- **Nikkei Weakly Aligned Sentences**
  /data/D4L/user/night/results/nikkei.007.low.snt 33518 pairs.

参考文献

[Al Onaizan et al 2000] Al Onaizan. Yaser et al, *Statistical Machine Translation, Final Report*, Johns Hopkins University, Baltimore, MD 2000.

[Bahl, Jelinek et al 1983] Bahl et al, *A Maximum Likelihood Approach to Continuous Speech Recognition, IEEE Transaction on Pattern Analysis and Machine Intelligence*, 5(2):179-190 1983.

[Brown, et al 1993] Brown et al, *The Mathematics of Machine Translation: Parameter Estimation, Computational Linguistics*, vol 19, number 2, pp 263-311, 1993.

[Cicekli and Guvenir 1996] Ciceklie, Ilyas, and Guvenir, H. Altay, *Learning Translation Templates from Bilingual Translation Examples*, NEMLAP 2, Proceedings of the 2nd International Conference on New Methods in Language Processing, Ankara, Turkey, pp 90-97, 1996.

[Charniak 2001] Charniak, Eugene, *A Maximum Entropy Inspired Parser*, In Proceedings of the North American Association for Computational Linguistics, 2000, ACL, New Brunswick, New Jersey.

[Clarkson, P and Rosenfeld, R 1997] Clarkson, P., and Rosenfeld, R., *Statistical Language Modelling Using the CMU-Cambridge Toolkit*, In Proceedings Eurospeech, 1997.

[Fung and Church 1993] Fung, P. and Church, K.W., *K-Vec: A New Approach for Aligning Parallel Texts*, In Proceedings 15th COLING pp 1096-1102 (1994).

[Gale and Church 1991] Gale, W.A. and Church, K.W., *A Program for Aligning Sentences in Bilingual Corpora*, ACL 1991 pp177-184.

[Haruno and Yamazaki 1996] Haruno, M. and Yamazaki, T., *High-Performance Bilingual Text Alignment Using Statistical and Dictionary Information*, ACL 1996 pp131-138.

[Kay, M 1981] Kay, M., *The Proper Place of Men and Machines in Language Translation*, Machine Translation 12;3–23, 1997.

[Kudoh and Matsumoto 2000] Kudoh, Taku. and Matsumoto, Yuji., *Japanese Dependency Structure Analysis Based on Support Vector Machines*, In Empirical Methods in Natural Language processing and Very Large Corpora, Pages 18–25, 2000.

[Manning and Schutze 1999] Manning, Chris. and Schutze, Hinrich., *Foundations of Statistical Natural Language Processing*, MIT Press 1999.

[McTait 2000] McTait, K., *Translation Patterns, Linguistic Knowledge and Complexity in an Approach to EBMT* Recent Advances in Example-Based Machine Translation, M. Carl and A. Way, (eds), Amsterdam, Kluwer Academic Press, forthcoming.

[Melamed 1998] Melamed, I.Dan., *Empirical Methods for Exploiting Parallel Texts* The MIT Press, Cambridge Massachussetts, 1998.

[Nightingale and Tanaka 2002] Nightingale, Stephen. and Tanaka, Hideki., *Aligning for SMT: Results from Real World Corpora*, Presented at the Forum on Information Technology, FIT, Tokyo 2002.

[Nightingale and Tanaka 2003] Nightingale, Stephen. and Tanaka, Hideki., *The Word is Mightier than the Count: Accumulating Translation Resources from Parsed Parallel Corpora*, in Proceedings CICLing 2003, Springer-Verlag LNCS 2588 pp420-431, Alexander Gelbukh (Ed).

[Och and Ney 2000] Och, Franz Josef. and Ney, Hermann., *Improved Statistical Alignment Models*, in Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, pp. 440-447, Hongkong, China, October 2000.

[Sumita and Iida 1991] Sumita, Eichiro, and Iida, Hitoshi., *Experiments and Prospects of Example Based Machine Translation*, in Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics, 1991.

[Tanaka et al 2002] Tanaka, Hideki. et al. *Speech to Speech Translation System for Monologues – Data Driven Approach* ICSLP, Denver Colorado, 2002.

[Veale, T and Way, A 1997] Veale, Tony and Way, Andy, *Gaijin: A Bootstrapping, Template-Driven Approach to Example-Based MT*, International Conference, Recent Advances in Natural Language Processing, Tzigov Chark, Bulgaria,pp 239-244, 1997.