An Enhancement Method for Japanese-English Automated Translation

Bartholomäus Wloka, Werner Winiwarter

Department of Scientific Computing
Vienna Austria
{bartholomaeus.wloka, werner.winiwarter}
@univie.ac.at

Abstract. We present a method for improving existing statistical machine translation methods using a knowledge base compiled from a bilingual corpus as well as sequence alignment and pattern matching techniques from the area of machine learning and bioinformatics. An alignment algorithm identifies similar sentences, which are then used to construct a better word order for the translation. Our preliminary test results indicate a significant improvement of the translation quality.

1 Introduction

Machine translation has been an active research area throughout the last 40 years. During this period, many promising concepts were proposed; however, there is still much room for improvement [34]. Especially when translating languages with radically different surface characteristics, as it is the case for Japanese-English, current machine translation techniques tend to produce unsatisfying results. The problems of automated translation between these languages become readily apparent when looking at current Web-based translations, e.g. from [9], which is shown in Fig. 1. While the translations of short phrases are of reasonable quality, translation systems struggle with long sentences. This is due to the growing complexity of sentences with increasing length and the vast differences in word and subclause order between these langauges. In general, the characteristics of the Japanese language pose a great challenge for translation into other languages [23, 21]. Those characteristics are:

- two syllabaries and a system of several thousand kanji, i.e. originally Chinese characters with several pronunciations and readings,
- lack of spaces to delimit word boundaries,
- a very high ambiguity in the grammar, as there exist no articles to indicate gender or definiteness,
- the tendency to omit information which can be inferred implicitly,
- sociolinguistic factors, e.g. avoiding direct and decisive expressions for reasons of politeness,
- an extensive system of formality with several levels of politeness forms, honorific expressions, and humble verb forms depending on the social status, relationship and other factors of the people involved.

Fig. 1. Example of current Web-based machine translation

To overcome those intricacies, we have directed our attention to a new and interdisciplinary approach. We have designed and implemented a method for finding structurally similar sentences with the help of an algorithm usually employed in the field of bioinformatics [17, 14]. This algorithm identifies sequence matches between chains of amino-acids, to find structurally similar proteins. The underlying assumption of our approach is that there is a significant overlap between the **structure** of a sentence and its **meaning**. In this article, we show that it is possible to enhance statistical machine translation results using this assumption. The *TRanslation Enhancement Framework* (TREF) [37] utilizes aligned and clustered sentence pair data to enhance the output of the statistical machine translation system *Moses* [12].

Though trained for the Japanese-English language pair, the system is modular and flexible. An adjustment or extension to other languages is a matter of changing mere implementation details and adding the language-specific resources, such as lexica, parser, corpora, etc. It is important to mention, however, that our translation framework is specifically designed and well-suited for languages with radically different surface characteristics, e.g. European-Asian language pairs.

The rest of this paper is organized as follows: In Sect. 2 the research relevant to our work is narrated, before we discuss TREF in Sect. 3. Section 4 presents our evaluation method and the results, followed by a conclusion and future work in Sect. 5.

2 Related Work

The ultimate goal of machine translation, i.e. abolishing language barriers, is presented by [28] in an entertaining narration. This ambitious pursuit of a system which will relieve the *lingua franca* and enable boundless communication between cultures is not quite yet in the realm of the possible. Nonetheless, research efforts towards this goal have been undertaken. In this section, we outline the research relevant to our work.

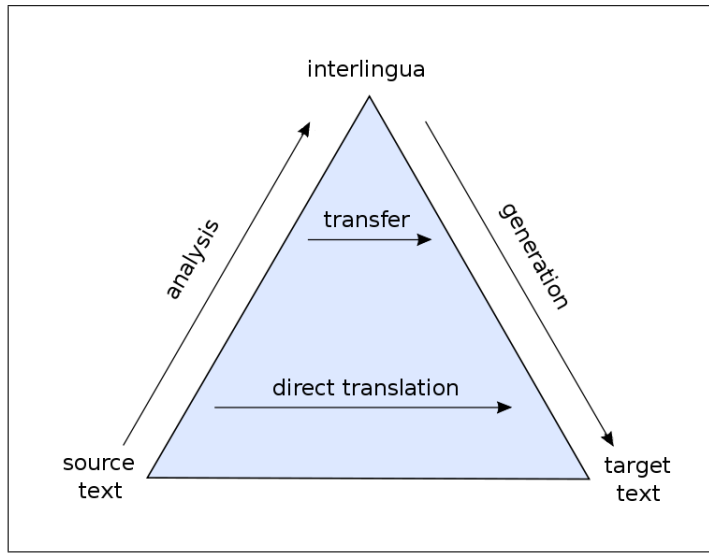

Fig. 2. Translation pyramid

2.1 Corpora

A vital resource for machine translation are bilingual corpora. Unfortunately, these are very rare, especially for the Japanese-English language pair. The currently predominant ones are the *Tanaka corpus* [31], the *Jenaad corpus* [32], and the *Verbmobil treebank* [10]. The Verbmobil treebank contains dialogs from telephone conversations in English, German, Japanese, and other languages, collected during the speech recognition research project of Verbmobil. The Japanese part contains around 160,000 words of text and is written in *Romaji*, i.e. the transcription of Japanese script into Roman literals. The Tanaka corpus consists of roughly 180,000 sentences and has a very broad domain. It has been collected over several years from various sources and compiled by Yasushito Tanaka in 2001. The Jenaad corpus is a collection of close to 150,000 sentence pairs. Extracted from news articles, it offers a certain consistency in terms of sentence types, while still offering a wide range of vocabulary and a variety of grammatical constructs. Because of these qualities, we have chosen the Jenaad corpus for our work. In addition, it is written in Japanese script, thereby avoiding potential ambiguities of the Romaji transcription.

2.2 Machine Translation

The research in machine translation has ever since included many different approaches. An overview of different techniques can be obtained from [34]. Their visual classification is exemplified by Vauquois' triangle in Fig. 2 [13].

The historically first method, located at the very top of the triangle, is the *interlingua* approach. It aims towards a language-independent representation, which mediates

between two or more languages. In contrast, *statistical machine translation* is at the bottom of the triangle, where no intermediate information is considered in the process, and there is a direct mapping from source to target text, depending on previously trained statistical data. A good overview of this technique can be obtained from [4].

Other approaches, which are also described in more detail in [1], are found somewhere between those two extremes, and the advantage of each depends on the demands of the given language pair. The challenges of translating Japanese to English gave birth to the new idea of *corpus-based* machine translation [25]. Apart from its success in translating between these languages, it further provides the opportunity for enhancing language learning environments by presenting the intermediate steps, i.e. the linguistic analysis of the translation process, to the learner. This was successfully accomplished by [35, 36]. The corpus-based method was quickly adopted by the machine translation community and merged with other techniques, as for example in [6]. Together with the idea of [24], that a mapping of grammatical functions and semantic roles is crucial for the Japanese-English pair, we have decided to mold these ideas into a new approach.

We have chosen a statistical machine translation method for a baseline translation in TREF, since it performs well in terms of translation of individual words and short phrases. It does not adhere to finding transition rules for syntax ordering and therefore represents a good first candidate for the post-editing done by TREF.

Amongst different tools, we have chosen *Moses*, since it is particularly effective when trained with a sufficiently large bilingual corpus. Moses scores well for structurally similar languages; however, for language pairs like Japanese-English, the word order is disarranged, which significantly lowers the quality of the translation, up to the point where the meaning of the sentence is irrecognizable. Moses does not consider any grammatical rules, so the output is syntactically wrong most of the time. The post-editing and rearranging of the Moses output aims at addressing this problem. Our method finds the correct word order for the translation result and produces a grammatically correct sentence, which conveys the meaning of its English counterpart. For a more detailed description of Moses, see Sect. 2.5

2.3 Natural Language Processing

To analyze the tokens of our bilingual corpus, we have used the *MontyTagger* from the *MontyLingua* project [19] for English, and *ChaSen* [22] for Japanese. Besides a part-of-speech tagging capability, MontyLingua offers an end-to-end natural language processing toolkit. ChaSen is a high-quality part-of-speech tagging tool for Japanese. Recently, *CaboCha* [18], a Japanese dependency parser which offers an even wider spectrum of NLP capabilities has been developed, and we plan to integrate it into TREF in the near future. In the following paragraphs, we give a brief overview of the NLP tools we have used in our work.

MontyLingua is a natural language processing engine written by [19], who based it on the work of [3]. It was programmed at the MIT Media Labs and written in Python. The source code is well structured and well documented, which simplifies utilizing it as a module. MontyLingua covers the full spectrum of text processing for the English

language, ranging from raw text processing to semantical analysis with summary generation. It is therefore referred to as an end-to-end natural language processing toolkit. It consists of six modules:

- *MontyTokenizer*: separation from punctuation (e.g. won't – wo n't)
- *MontyTagger*: POS tagging with "common sence"
- *MontyLemmatiser*: POS sensitive lemmatisation (plural removal, infinitive form)
- *MontyREChunker*: separates tagged text into verb, noun and adjective
- *MontyExtractor*: extracts various semantically valuable information from sentences
- *MontyNLGenerator*: generation of reports

The MontyTokenizer module separates words from punctuation marks, thereby creating tokens. The separation is not done for acronyms, as far as they are classified correctly. For example:

children's	children 's
students'	students '
won't	wo n't
he's	he 's
U.S.A.	U.S.A.

After tokenizing the words, the MontyTagger module identifies the tag of each token.

Japan	Japan/NNP
cat	cat/NN
had	had/VBN
he's	he/PRP 's/VBZ

The "common sense" functionality in the MontyTagger module from the ConceptNet project[19, 11] extracts semantic information from sentences, and enables the improvement of accuracy of the POS tagging process. For example, by identifying subject/verb/object tuples, it avoids tagging errors as seen in Fig. 3, where the common sense identifies the falsely classified word "bit" as a noun phrase and corrects it to verb phrase.

Once the text was processed with the tokenizer and the tagger, the MontyLemmatiser can be used to find the *lemma* of a token. A lemma, in linguistics, is the canonical form of a set of forms, which are in this case the infinitive forms of verbs, the singular forms of nouns in the plural form, etc.

For example:

cats	cats/NP/cat
had	had/VBN/have
words	words/NNS/word

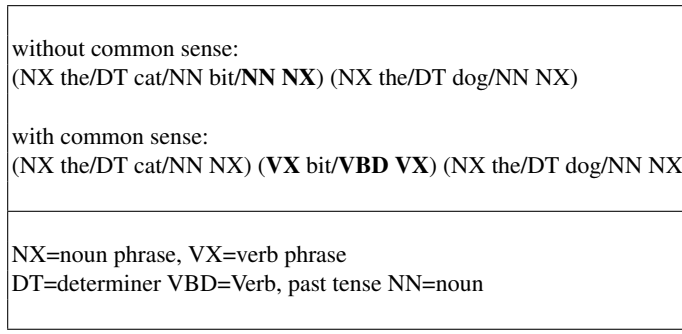

Fig. 3. "Common sense" of MontyTagger

The MontyREChunker module, which is executed via MontyExtractor, is a shallow parser, i.e. a pre-step for syntactical analysis and semantic interpretation. The MontyNLGenerator generates reports with the information gathered throughout the process. An example output of the digest, created by the MontyNLGenerator, is depicted in Fig. 4.

ChaSen is a morphological parser for the Japanese Language. It was developed at the Matsumoto laboratory, Nara Institute of Science and Technology. This work was based on the Japanese morphological analyzer *Juman*, version 2.0. As stated in [22], the biggest problem that challenged the creators of ChaSen was the lack of a commonly accepted grammar and of a consolidated/joint grammatical terminology. Even though word classifications and some grammatical terminology are taught in schools, researchers do not hold these in high regard. Furthermore, they are not suitable for computer processing. Hence, ChaSen differs from NLP tools for other languages in the point that it does not categorize according to linguistically defined categories but rather by practically encountered patterns in the language. Due to this fact, it was impossible for us to find a list with English translations of the tags and morphosyntactical categories used by ChaSen. Therefore, we have compiled a translation ourselves. It is important to note that this list is translated from the European point of view to improve the understanding for the English speaking audience.

We describe the analysis of ChaSen by giving an example of one output line. The following analyzed token, the verb する(suru), means "to do": し シ する 47 3 7. The first part is the token as it appears in the analyzed sentence. In this case it is "し" (shi), the form of the verb "suru", as it is changed before a particle which indicates the past tense. The following information "シ" is the token written in katakana script, followed by the dictionary form of the verb. The first of the three numbers at the end of the output is the tag information, 47 being an independent verb, and the last two numbers indicating the morphosyntactical information, in this case the conjugation to past tense.

```
> it's not clear whether he will show up

(NX it/PRP NX) (VX 's/VBZ not/RB VX) (NX clear/JJ NX)
whether/IN  (NX he/PRP NX) (VX will/MD show/VB VX) up/IN

   SENTENCE #1 DIGEST:
         adj_phrases: ['clear']
  adj_phrases_tagged: ['clear/JJ']
           modifiers: ['not', 'clear']
    modifiers_tagged: ['not/RB', 'clear/JJ']
        noun_phrases: ['it', 'he']
 noun_phrases_tagged: ['it/PRP', 'he/PRP']
parameterized_predicates: [[['not ', ['negation',
'past_tense']],  ['it', []], ['clear', []], ['whether he',
['prep=whether']]], [['show', []], ['he', []]]]
        prep_phrases: ['whether he']
 prep_phrases_tagged: ['whether/IN he/PRP']
 verb_arg_structures: [["'s/VBZ not/RB", 'it/PRP',
 ['clear/JJ', 'whether/IN he/PRP']], ['will/MD show/VB',
 'he/PRP', []]]
verb_arg_structures_concise: ['("not " "it" "clear" "whether
 he")', '("show" "he" )']
        verb_phrases: ["'s not", 'show']
 verb_phrases_tagged: ["'s/VBZ not/RB", 'show/VB']
None
[['not ', 'it', 'clear', 'whether he'], ['show', 'he']]

GENERATED SUMMARY:
It not  clear whether he showed .
-- monty took 0.01 seconds. --
```

Fig. 4. MontyLinguaNLGenerator output

2.4 Sequence Alignment

The Needleman-Wunsch algorithm for computing similarities in protein building blocks, i.e. amino-acid chains, was published in 1970 [26]. Quickly, many derivatives and extensions of this method followed. The basic idea behind this concept was to depict amino-acid chains as strings of alphabetic characters, align them to offer the best match between two strings, and compute a similarity measure [30]. This method was further improved by [15], using a distance measure in conjunction with dynamic programming. Many other research efforts found different distance measures to identify the similarity of sequences. The approach of [16] is generic enough to be extended to the area of machine translation, therefore we use it in our research effort by treating sentences from a bilingual corpus analogously to the sequence alignment of amino acid chains.

Input Sentence 1:	エリツィン大統領の指導の下で、ロシア政府は、困難な改革過程に乗り出した。
Tagged Moses:	JJ/ NN/ NN/ NN/ IN/ DT/ NN/ IN/ DT/ JJ/ NN/ IN/ DT/ NN/ NN
Correct Translation:	Under President Yeltsin's leadership the Russian government has embarked on a difficult reform process.
Moses Translation:	russian president boris yeltsin under the leadership of the russian government on the reform process .
Input Sentence 2:	不平等条約だと批判される理由だ。
Tagged Moses:	PRP/ VBZ/ JJ/ NN/ VBN
Correct Translation:	This is a reason why the treaty is being criticized as unequal.
Moses Translation:	it is unequal treaty criticized .

Fig. 5. Moses output example

2.5 Moses – Statistical Translation

Moses is a very widely used translation tool for multiple languages. It uses a statistical machine translation approach [12]. After the input of an extensive amount of training data, it yields decent translation results; however, the examples below show that translations can be quite wrong. Grammatically complicated translations are very likely to be incorrect, as can be seen in Fig. 5.

Moses uses the concept of phrase-based translation. Each sentence is segmented into word sequences or *phrases* and is then translated into a phrase in the target language, which can be reordered. In order to formulate the translation probability, the Bayes formula is used

$$\text{argmax}_e p(e|f) = \text{argmax}_e p(f|e)p(e)$$

It separates the language model $p(e)$ and the translation model $p(f|e)$. In this example, e stands for the English language and f for the foreign language. The best English translation is obtained with

$$e_{\text{best}} = \text{argmax}_e p(f|e)p_{LM}(e)\omega^{\text{length}(e)}$$

$p(f|e)$ being the language model defined by

$$p(\bar{f}_1^I|\bar{e}_1^I) = \Phi_{i=1}^I \phi(\bar{f}_i|\bar{e}_i)d(\text{start}_i, \text{end}_{i-1})$$

where $d(start_i, end_{i-1})$, is the relative distortion probability which models the reordering and $\phi(\bar{f}_i|\bar{e}_i)$ is the probability distribution for the translation. In the distortion model, $start_i$ denotes the start position of the foreign phrase that was translated into the i^{th} phrase, and end_{i-1} denotes the end position of the foreign phrase, which was translated into the $(i-1)^{th}$ English phrase. In order to calibrate the output length, a word cost factor ω is added for each generated English word. p_{LM} is a trigram language model representing a simple optimizing factor.

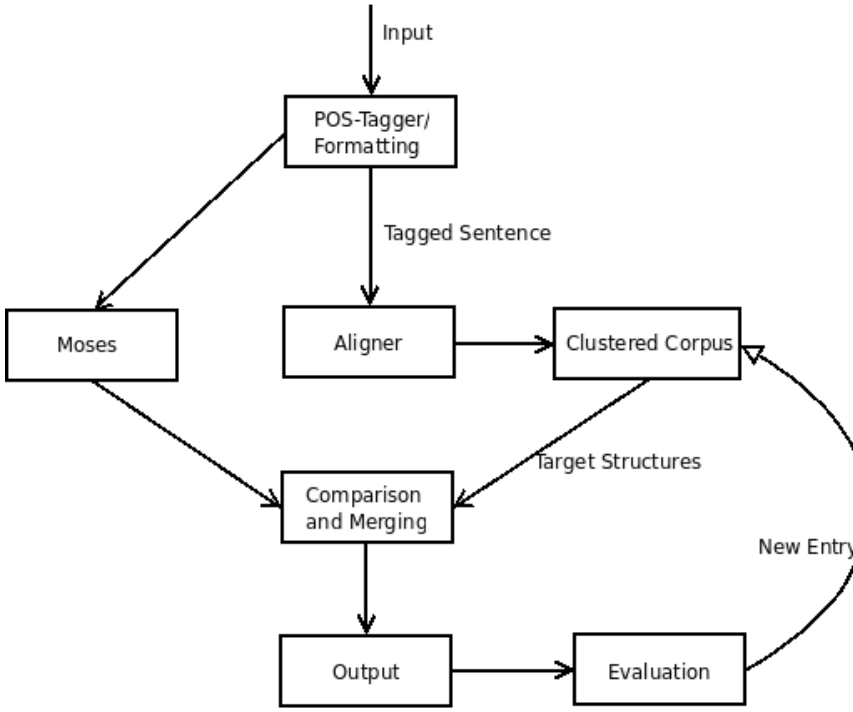

Fig. 6. Overview of dataflow

The phrase probability translation table and the reordering table are obtained by aligning the words in a bilingual corpus using *GIZA++* [27], a toolkit implementation of the original IBM Models that started statistical translation. Once the phrase table is built, an n-gram table is constructed by Moses, which is used to identify n-grams in new input sentences. A training process of a reordering table can take a long time, depending on the size of the corpus and the size of n. We have split the Jenaad corpus in a training set of 149,000 sentences and reserved 1,000 sentences for testing. The resulting reordering table, which was trained with trigrams, contains over 44 Million characters in approximately 15 Million lines. The phrase table has 131 Million character entries in almost 1 Million lines.

3 TREF

The overview of the architecture of TREF is shown in Fig. 6. The *PoS- Tagger/Formatting* module tokenizes the input sentence and assigns PoS tags in a format which is described below. The sentences in their tokenized format are then aligned with the clustered cor-

石炭の利用拡大は大気汚染をさらに悪化させる sekitan no ryou kakudai wa taiki osen wo sarani okka saseru		
石炭/セキタン/2/0/0	の/ノ/71/0/0	利用/リョウ/17/0/0
拡大/カクダイ/17/0/0	は/ハ/65/0/0	大気/タイキ/2/0/0
汚染/オセン/17/0/0	を/ヲ/61/0/0	さらに/サラニ/56/0/0
悪化/アッカ/17/0/0	さ/サ/47/3/5	せる/セル/49/6/1

Fig. 7. Tagged Japanese sentence

expanded use of coal worsens air pollution						
expanded VBN	use NN	of IN	coal NN	worsens VBZ	air NN	pollution NN

Fig. 8. Tagged English sentence

pus to find the target structure, which is sent to the *Comparison and Merging* module. This module takes this input as well as the translation from *Moses* and enhances its translation quality by applying a template approach. The resulting translation can then be evaluated and added to the corpus. Each step is described in detail in the following subsections.

3.1 Part-of-Speech Tagging

The input sentence is sent to either one of the *part-of-speech* (PoS) tagging modules MontyTagger [20] or ChaSen [32]. The result of this process can be seen in Fig. 7 and Fig. 8 for Japanese and English respectively. The Japanese sentence is written in Roman transcription for the reader's convenience. The tags produced by ChaSen consist of a sentence token, its *katakana* representation (one of the Japanese syllabaries, which indicates the pronunciation of a kanji), and a numerical representation of the morphological data. The English tags contain the word itself and the PoS tag as an acronym.

After each sentence token is assigned a PoS tag, the sentence and its tags are compared with the sentences already stored in a clustered corpus, which is a customized and enriched version of the *Jenaad* Corpus [32]. We have modified it by removing as much noise as possible, assigned PoS tags to each sentence token, and stored them in an SQL database. Additionally, we have implemented a post-processing step for the PoS-tagging to correct mistakes by the MontyTagger, such as the wrong tagging of words written in capital letters, e.g. at the beginning of a sentence. We have kept the data with all available PoS tags and additionally created a reduced and optimized tag set, which provides a quick access for efficient processing. Other representations and tag sets can be added easily to satisfy different needs in future work.

d(nn(house),nn(house))	= 0
d(nn(house),nn(office))	= 0.5
d(nn(house),dt(the))	= 1

Fig. 9. Distance calculation example

S1	He	went	to	the	store	to	buy	(g)	milk
T1	PRP	VBD	TO	DT	NN	TO	VB	(g)	NN
S2	She	hurried	to	the	university	to	attend	a	lecture
T2	PRP	VBD	TO	DT	NN	TO	VB	DT	NN
D	0.5	0.5	0	0	0.5	0	0.5	1	0.5
$SentenceDistance$	$\frac{1}{2 \times 9} \times (0.5+0.5+0+0+0.5+0+0.5+1+0.5) = 0.19444$								

Fig. 10. Sequence alignment distance calculation example

3.2 Aligning and Clustering

In order to identify similar sentences, we have used a slightly modified alignment algorithm from bioinformatics. Instead of aligning protein chains, we align chains of words, i.e. sentences. We have applied relational sequence alignment [17, 14] to obtain clusters of structurally similar sentences. The alignment is done according to the Nienhuys-Cheng distance function.

An example of a distance between the tokens of each sentence is shown in Fig. 9. If the token and its PoS tag differ, the distance is 1. In the case of a structural match, the distance is 0.5, and 0 for a perfect match. The subsequent distance calculation of an entire sentence is depicted in Fig. 10. Gaps, which are identified and symbolized with *(g)* in the example, are assigned variable *gap penalties*. In order to achieve better matching results, we differentiate between *gap opening* and *gap extension*, which allows us to separate subordinate clauses from otherwise non-matching word sequences.

Applying the distance function to English and Japanese sentences results in two cluster sets, which can be utilized for sentence matching and translation as described below. The size of the clusters can be directly altered by a threshold value on the distance between sentences. Finding the parameters of the distance function as well as a suitable threshold value for clustering is done empirically and is an important part of the entire system, since the resulting similar sentences rely on these parameters, which, in turn, define the quality of the translation improvement. These similarity measure parameters can be adjusted to fine-tune the result, depending on the text type and text domain. By allowing lower similarity values, a higher number of candidates can be produced, whereas a higher similarity value reduces the number of candidates. This flexibility can be utilized for a language learning application to present an arbitrary amount of similar translations to the student. The output is then evaluated by the user and added to the corpus. Once the distances are computed, clusters can be defined by setting a threshold value. This concept is shown in Fig. 11 in a Cartesian coordinate system. Each sentence

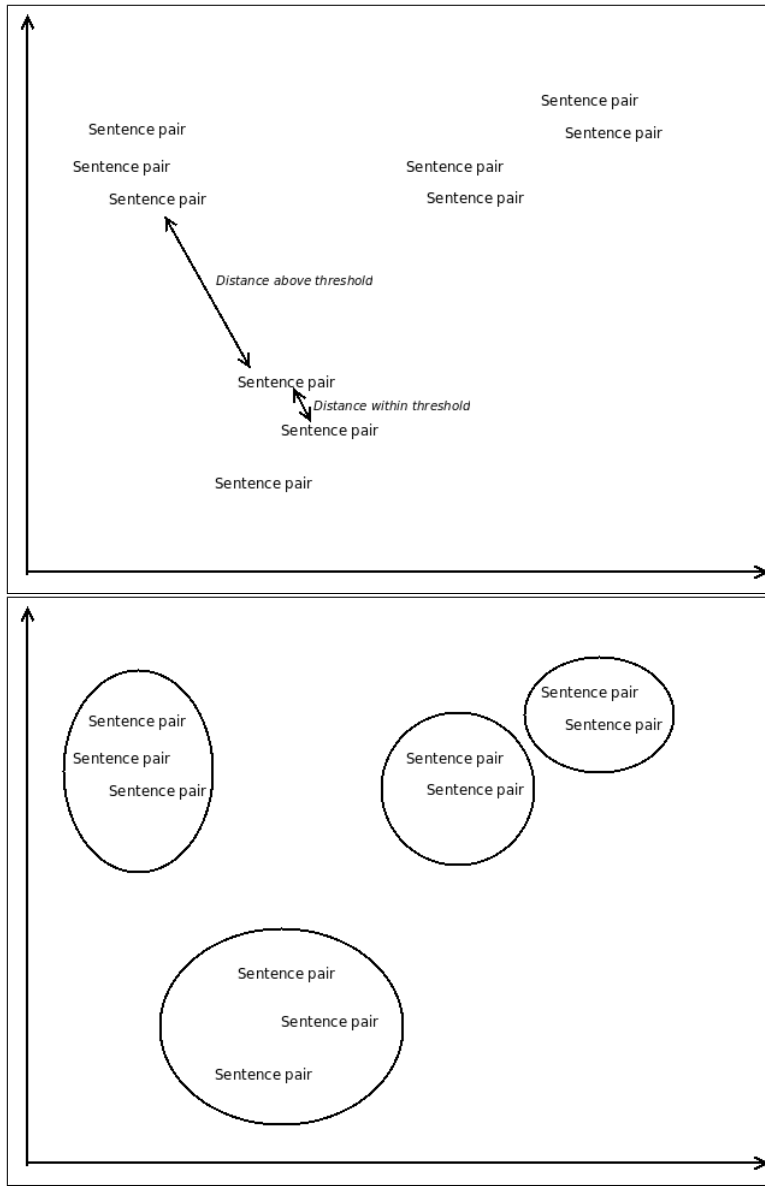

Fig. 11. Clusters in Euclidean space

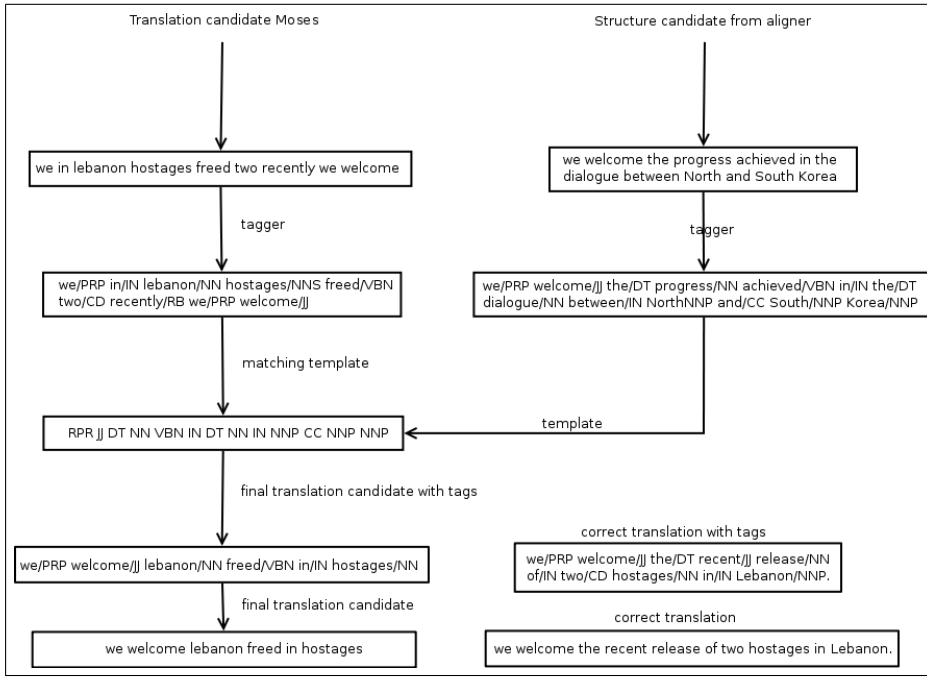

Fig. 12. Workflow example

which has a distance lower than a certain threshold value, is assigned to a cluster and is therefore considered *structurally similar* to sentences in this cluster.

3.3 Comparison and Merging

The comparison of the query sentence with the clusters yields several similar structures. At the same time, the query sentence is processed with Moses to obtain a preliminary

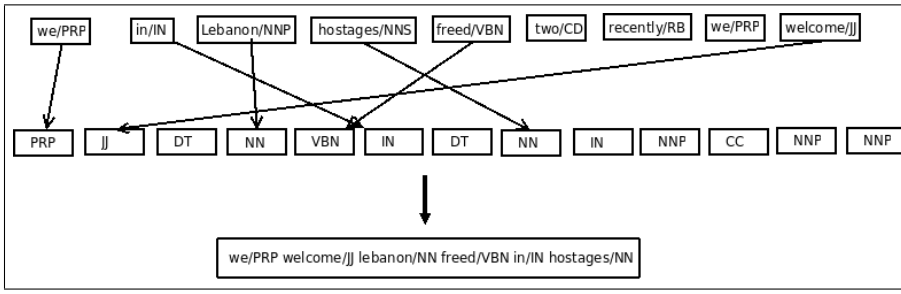

Fig. 13. Matching

translation. This translation is then used to fill the template of the structures which have been found in the previous step. Thereby, a certain number of translation candidates is produced. The result of the procedure including all intermediate results is depicted in Fig. 12.

For the purpose of demonstration, we use the Japanese translation of the short input sentence: “We welcome the progress achieved in the dialog between North and South Korea.”. On the left side, the translation candidate proposed by Moses is tagged and prepared for the matching step. On the right side, the output from the aligning step, i.e. the translation of the sentence structurally similar to the Japanese input, is also tagged and is transformed into a template by removing everything except the PoS-tags. Finally, the template is filled with the translation candidate from Moses to produce the improved translation. The filling of the structure templates from the aligning step is shown in Fig. 13 in greater detail. The translation by Moses is: “we in Lebanon hostages freed two recently we welcome”. TREF transforms this, by filling the structure template into “we welcome Lebanon freed in hostages”. It is apparent that some tokens are lost in the process of filling the template, which leaves room for future work and potential for further improvement of the translation quality.

3.4 Web Interface

The clustered corpus of PoS-tagged sentence tokens in several representations as well as morphological information, is stored in a MySQL database and is accessible through a Django Web framework [7]. In Django, all interactive content as well as settings, modules, and database setup are written in Python, which made it a good candidate for our system due to its powerful string and text manipulation capabilities. Further, Django provides stable Web development and administrative utilities. In particular, the communication to the database and efficient Web design tools including HTML code inheritance made it an ideal developing environment. The structure of the framework is depicted in Fig. 14.

From the main site, the user can navigate to the translation module, the sentence pair input, the random sentence output, as well as legends for the PoS tags for English and Japanese. The translation module offers an interface which, upon input of a sentence, sends it to the server and – after the above described translation process – displays the result. The sentence input module takes a sentence pair input, which is flagged as a new addition and is checked manually before being added to the database. The random sentence output is a first step towards the language learning functionality and outputs a sentence from the database including its translation, its tags, and morphological information. We have created a page for the explanation of PoS tags. The translation of the original Japanese ChaSen tags into English is, to the best of our knowledge, the only English ChaSen PoS-tag legend available. The framework is available on the Web server maintained by the authors [38].

3.5 Showcase

Figure 15 shows an example of the workflow from the input of a sentence to an output of several translation candidates. The input ”My name is Yamada.” is tagged and com-

Fig. 14. Structure of the Web framework

pared with the clustered data. The PoS tags for the sentence in this case are: `My/POP` (personal pronoun), `name/NN` (noun), `is/VBZ` (verb), `Yamada/NNP` (proper noun). The alignment detects sentences in the database, that are similar in terms of words and PoS-tags (see Fig. 9). The translations of the identified structures are also checked for similarities with other clusters. This step, which we call *structure-to-meaning-mapping*, identifies other structures of potential translation candidates. According to the threshold settings and a maximum value, which defines the number of sentences which are extracted from the database, a certain number of structures are handed down to the next step. These structures are sent to the *matching and translation step*, where the structures and the output from Moses are merged to yield the final output, i.e. the translation candidates. Figure 16 shows the results for the two example sentences from Fig. 5.

4 Evaluation

To create a testing scenario, we have extracted 1,000 out of the total 150,000 sentences from the Jenaad corpus. The remaining 149,000 sentences were used as training data for Moses and for clustering. Due to the long processing time for each sentence, we

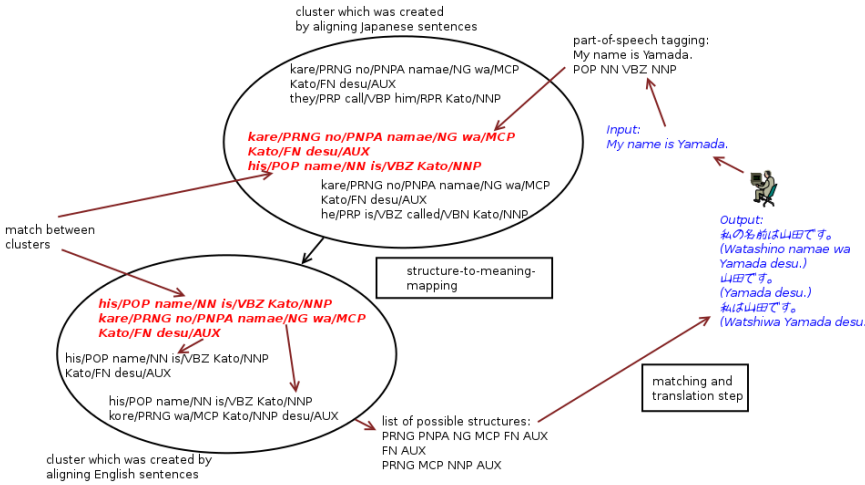

Fig. 15. Translation via Clustering

Input Sentence 1:	エリツィン大統領の指導の下で、ロシア政府は、困難な改革過程に乗り出した。
Tagged Moses:	JJ/ NN/ NN/ NN/ IN/ DT/ NN/ IN/ DT/ JJ/ NN/ IN/ DT/ NN/ NN
Aligner Output:	With the Soviet Communist Party's Central Committee now discussing abandoning one-party rule, democratization has entered a significant stage, and U.S.-Soviet detente is progressing well.
Tagged Aligner:	IN/ DT/ NNP/ NNP/ NNP/ POS/ NNP/ NNP/ RB/ VBG/ VBG/ JJ/ NN/ ,/ NN/ VBZ/ VBN/ DT/ JJ/ NN/ ,/ CC/ NNP/ NNP/ ./ JJ/ NN/ VBZ/ VBG/ RB/ ./
Correct Translation:	Under President Yeltsin's leadership the Russian government has embarked on a difficult reform process.
Moses Translation:	russian president boris yeltsin under the leadership of the russian government on the reform process .
Improved Structure:	under the president boris yeltsin leadership government russian reform process the russian .
Input Sentence 2:	不平等条約だと批判される理由だ。
Tagged Moses:	PRP/ VBZ/ JJ/ NN/ VBN
Aligner Output:	Hostage-taking should be denounced as international terrorism.
Tagged Aligner:	NNP/ MD/ VBN/ IN/ JJ/ NN/
Correct Translation:	This is a reason why the treaty is being criticized as unequal.
Moses Translation:	it is unequal treaty criticized .
Improved Structure:	treaty is criticized unequal .

Fig. 16. Test run example of Moses

Sentenc Nr.	TREF	Moses	signed rank	Sentenc Nr.	TREF	Moses	signed rank
1	20	15	+6.5	21	65	75	-15.5
2	40	50	-15.5	22	65	35	+32.5
3	30	30	—	23	20	25	-6.5
4	25	25	—	24	10	20	-15.5
5	30	35	-6.5	25	0	0	—
6	50	30	+26.5	26	50	45	+6.5
7	45	30	+21.5	27	15	35	-26.5
8	60	30	+32.5	28	20	15	+6.5
9	25	40	-21.5	29	40	50	-15.5
10	50	25	+29.5	30	30	30	—
11	50	35	+21.5	31	25	25	—
12	25	20	+6.5	32	30	35	-6.5
13	10	15	-6.5	33	50	30	+26.5
14	65	75	-15.5	34	45	30	+21.5
15	65	35	+32.5	35	60	30	+32.5
16	20	25	-6.5	36	10	20	-15.5
17	50	25	+29.5	37	0	0	—
18	50	35	+21.5	38	50	45	+6.5
19	25	20	+6.5	39	15	35	-26.5
20	10	15	-6.5	40	25	40	
W=139, $n_{s/r}$=34, z=1.18, P(1-tail)=0.119							

Fig. 17. Wilcoxon signed rank test

have decided to analyze fewer sentences in detail instead of using standard scoring tools, such as [29] or [8], which would be more significant for larger amounts of output. Moreover, the validity of automated scoring tools of this kind has been criticized by [1, 5]. Hence, our evaluation was done by an expert who judged each translation on four categories: word order, word translations, semantics, and fluency. The categories were equally weighted with a top score of 25 each (see Fig. 18). A total of 40 sample sentences were evaluated, and a statistical significance of the result was verified with a Wilcoxon signed-rank test, depicted in Fig. 17 [33]. The result was a better score for the sentences processed with TREF with a score of W=139 over a sample size of N=34 and a P(1-tail) value of 0.119.

5 Conclusion

In this article, we have described a design for enhancing state-of-the-art machine translation using sequence alignment from the area of bioinformatics, combined with PoS tagging and clustering of a bilingual corpus. Our results have proven that similarities in sentence structure can be used to create templates for translation candidates, in particular for the Japanese-English language pair. We have described our implementation of the system and its Web framework. We have trained the system with the Jenaad Corpus and tested the system for Japanese-English. The evaluation of the system yielded promising results. At the time of writing, TREF is already integrated in another re-

Input sentence:	我々は、レバノンにおける復興努力を支持する。			
Correct Translation:	We support the efforts of reconstruction in Lebanon.			
Moses Translation:	we support in lebanon reconstruction efforts.			
Enhanced by TREF:	we support lebanon in reconstruction.			
	Word Order	Word Translations	Semantics	Fluency
Moses:	5	15	15	5
TREF:	15	15	20	15
Total Score Moses: 40 Total Score TREF: 65				

Fig. 18. Example evaluation

search project focusing on ubiquitous translation and language learning with the help of mobile devices.

For future work, we plan to optimize the parameters in the aligning process to fine-tune the word reordering as well as adding grammatical parsing steps after the template filling to improve the syntactical correctness of the sentence. An additional dictionary lookup will be integrated to amend word translations, which could not be processed by the statistical translation step.

We want to extend the language learning aspect of the system to offer a Web-based learning platform and improve the efficiency of the entire system with pre-computing and indexing methods. We plan to incorporate a Japanese dependency parser. The currently active research efforts on the Japanese WordNet [2] and CaboCha [18] are promising candidates for an additional extension of TREF as a language learning platform offering extensive semantic and syntactic information as well as visual representations of vocabulary.

References

1. Christian Boitet, Herve Blanchon, Mark Seligman, Valerie Bellynck. Evolution of MT with the Web: In: Proceedings of the Conference ”Machine Translation 25 Years On”: Cranfield, England (2009)
2. Francis Bond et al.: Enhancing the Japanese WordNet. In: Proceedings of the 7th Workshop on Asian Language Resources, in conjunction with ACL-IJCNLP. (2009)
3. Eric Brill: Transformation-based error-driven learning, natural language processing: a case study in part-of-speech tagging: Comput. Linguist, MIT Press, 21, 4, 543–565, (1995)
4. Peter Brown et al.: A statistical approach to machine translation, Comput. Linguist. MIT Press 16, 2, 79–85 (1990)
5. Chris Callison-Burch, Miles Osborne: Re-evaluating the role of BLEU in machine translation research. In: Proceedings of the Conference EACL, pp. 249–256.(2006)
6. Michael Carl, Andy Way, Walter Daelemans: Recent Advances in Example-Based Machine Translation: Comput. Linguist. MIT Press, 30, 4, 516–520 (2004)
7. Django Project `http://www.djangoproject.com`

8. G. Doddington: Automatic evaluation of machine translation quality using n-gram co-occurrency statistics. In: Proceedings of the ARPA Workshop of Human Language Technology. (2002)
9. Excite Japan, Translation www.excite.co.jp/world/english
10. M. Finke, P. Geutner, H. Hild, T. Kemp, K. Ries, M. Westphal: The Karlsruhe-Verbmobil Speech Recognition Engine. Acoustics, Speech,, Signal Processing: IEEE Computer Society 1, 83 (1997)
11. Catherine Havasi and Rob Speer and Jason Alonso: ConceptNet 3:a Flexible, Multilingual Semantic Network for Common Sense Knowledge. In: Proceedings of Recent Advances in Natural Language Processing. (2007)
12. Hieu Hoang, Alexandra Birch, Chris Callison-burch, Richard Zens, Rwth Aachen, Alexandra Constantin, Marcello Federico, Nicola Bertoldi, Chris Dyer, Brooke Cowan, Wade Shen, Christine Moran, Ondrej Bojar: Moses: Open source toolkit for statistical machine translation, pp. 177–180. Association for Computational Linguistics (2007)
13. W. John Hutchins, Harold L. Somers: An Introduction to Machine Translation: Academic Press (1992)
14. Andreas Karwath, Kristian Kersting: Relational Sequence Alignments, Logos. In: Inductive Logic Programming 16th International Conference, pp. 290–304. Springer (2007)
15. Andreas Karwath, Kristian Kersting: Relational Sequence Alignments. In: Proceedings of the 4th International Workshop on Mining, Learning with Graphs (MLG'06) (2006)
16. Andreas Karwath, Kristian Kersting, Niels Landwehr: Boosting Relational Sequence Alignments. In: Proceedings of the 8th IEEE International Conference on Data Mining (2008)
17. Kristian Kersting, Luc De Raedt, Bernd Gutman, Andreas Karwath, Niels Landwehr: Probabilistic Inductive Logic Programming: Relational Sequence Learning: Springer Berlin/Heidelberg (2008)
18. Taku Kudo, Yuji Matsumoto: Japanese Dependency Analysis using Cascaded Chunking: In: CoNLL 2002: Proceedings of the 6th Conference on Natural Language Learning 2002 (COLING 2002 Post-Conference Workshops), pp. 63-69 (2002)
19. H. Liu: An end-to-end natural language processor with common sense: MIT Media Lab (2004)
20. H. Liu, P. Singh: ConceptNet — A Practical Commonsense Reasoning Tool-Kit: BT Technology Journal, Kluwer Academic Publishers 22, 4, 211–226 (2004)
21. S. Makino, M. Tsutsui: A Dictionary of Basic Japanese Grammar: The Japan Times (1986)
22. Yuji Matsumoto, Akira Kitauchi, Tatsuo Yamashitaka Hirano, Hiroshi Matsuda, Kazuma Takaoka, Masayuki Asahara: Japanese Morphological Analysis System ChaSen version 2.2.1: Users Manual (2000)
23. Y.M. McClain: Handbook of Modern Japanese Grammar: The Hokuseido Press (1981)
24. Teruko Mitamura, Nyberg Eric: Hierarchical Lexical Structure, Interpretive Mapping in Machine Translation. In: Proceedings of the 14th Conference on Computational Linguistics, pp. 1254–1258. Association for Computational Linguistics (1992)
25. Makato Nagao: A framework of a mechanical translation between Japanese, English by analogy principle. In: Proceedings of the international NATO symposium on Artificial, human intelligence, pp. 173–180 Elsevier North-Holland, Inc. (1984)
26. S.B. Needleman, C.D. Wunsch: A general method applicable to the search for similarities in the amino acid sequence of two proteins: Journal of Molecular Biology 48, 2, 443-453 (1970)
27. Franz Josef Och, Hermann Ney: A Systematic Comparison of Various Statistical Alignment Models: Computational Linguistics, 1, 29, 19–51, (2003)
28. Nicholas Ostler: The Jungle Is Neutral – Newcomer Languages Face New Media. In: Proceedings of the 13th Annual Conference of the European Association for Machine Translation, University Politecnica de Catalunya Barcelona Spain (2009)

29. Kishore Papineni, Salim Roukos, Todd Ward, Wei-Jing Zhu: BLEU: a method for automatic evaluation of machine translation. In: ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 311–318. Association for Computational Linguistics (2002)
30. T.F. Smith, M.S. Waterman: Identification of common molecular subsequences: Journal of Molecular Biology, 147, 195–197 (1981)
31. Yasushito Tanaka: Compilation of a multilingual parallel corpus. In: Proceedings of the PACLING 2001, pp. 265–268. Kyushu, Japan (2001)
32. Masao Utiyama, Hitoshi Isahara: Reliable measures for aligning Japanese-English news articles, sentences. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, pp. 72–79. Association for Computational Linguistics (2003)
33. Frank Wilcoxon: Individual Comparisons by Ranking Methods: Biometrics Bulletin, 1, 6, 80–83 (1945)
34. Yorick Wilks: Machine Translation: Its Scope and Limits. Springer (2008)
35. Werner Winiwarter: WILLIE – a Web Interface for a Language Learning, Instruction Environment. In: Proceedings of the 6th International Conference on Web-based Learning: Springer (2008)
36. Werner Winiwarter: WETCAT – Web-Enabled Translation Using Corpus-Based Acquisition of Transfer Rules. In: Proceedings of the Third IEEE International Conference on Innovations in Information Technology: Dubai, United Arab Emirates (2006)
37. Bartholomäus Wloka: Enhancing Japanese-English Machine Translation – A Hybrid Approach: University of Freiburg (2009)
38. Bartholomäus Wloka, Werner Winiwarter Project Web-site `https://wloka.dac.univie.ac.at/project/`