文章编号：1003-0077（2011）00-0000-00

专利文本统计机器翻译中使用微粒群优化改进 Champollion*

熊文[1,2]，蒋宏飞[1,2]，任智军[1]，姜涛[1]，张凯[1]

（1.国家知识产权局中国专利信息中心，北京 100875；2. 北京师范大学中文信息处理研究所，北京 100875）

摘要：研究了潜在噪声并行文本句子对齐鲁棒算法 Champollion。针对专利文本统计机器翻译，提出基于群智的微粒群优化（PSO，particle swarm optimization）改进 Champollion 对齐精度。用三个优化权重改写 Champollion 相似度函数，以较小代价搜索最优对齐惩罚、长度惩罚参数。方法在训练集上用 PSO 生成参数和权重，调用 Champollion；用 F1 值评估句子对齐结果；将评估值作为微粒适应度反馈到 PSO；通过 PSO 发现最优参数和权重。在 6 个通用数据集和 1 个专利集上实验结果表明， Champollion 精度得到提高，验证了方法有效性。

关键词：机器翻译；微粒群优化；句对齐

中图分类号：TP391　　**文献标识码：**A

Improving Champollion using PSO in SMT for Patent texts

Wen Xiong[1,2], HongFei Jiang[1,2], ZhiJun Ren[1], Tao Jiang[1], Kai Zhang[1]

(1. The China Patent Information Center, the State Intellectual Property Office, Beijing 100875, China ; 2. Beijing Normal University, Institute of Chinese Information Processing, Beijing 100875, China)

Abstract: To investigate a robust sentence aligner for potential noisy parallel text as Champollion algorithm, this paper presents a method to improve the precision of sentence alignment of Champollion using particle swarm optimization (PSO) based on swarm intelligence for patent parallel texts in statistical machine translation. The method reforms the similarity function in Champollion using three optimized weights, and search optimal alignment penalty parameters and length penalty parameters with low cost. First, the method calls Champollion using the parameters and weights generated by PSO on training data set. Then, it evaluates the results of sentence alignment using the F1 value. Next, it feeds the evaluation back to the PSO as the fitness of particles. Finally, the optimal parameters and weights are found by PSO. The experimental results show the precision of Champollion on six general data sets and one patent data set is improved, which verifies the effectiveness of the method.

Key words: Machine translation；Particle Swarm Optimization；Sentence Alignment

1 Introduction

Parallel text has an important value for statistical machine translation (SMT) [1]. Sentence alignment is an important step in SMT for utilizing it, which aligns sentences of source language with sentences of target language to pair. After that, a word alignment and phrase translation pairs are extracted.

However, it is only effective to align sentences manually for small-scale parallel texts, which provides training data set for automatic sentence alignment for large-scale parallel texts. For aligning sentences automatically, there are many

* **收稿日期：**2013-10-08　**定稿日期：**2013-10-13

基金项目：Hi-Tech Research and Development Program of China (2012AA011104), and Postdoctoral Science Foundation of China (Grant No.: 2013M540125, 2013M530026).

作者简介：熊文（1968——），男，助理研究员/博士后，机器翻译，数据挖掘，文本挖掘；蒋宏飞（1982——），男，助理研究员，自然语言处理，机器翻译，专利信息处理；任智军（1977——），男，高级工程师，机器学习，机器翻译。

methods appeared in literatures, such as sentence length based method, lexicon-based method, and hybrid method. Those methods can effectively handle the problem in the close-language pair, but their performance declines fast in the remote-language pair. For example, the performance in English-French pair is higher than that in English-Chinese pair.

On the other hand, some methods cannot handle effectively the parallel texts with noise, which is introduced in the extraction from the different format of documents, such as PDF, WORD, and WordPerfect. Thus, Champollion [2] was presented, which is a robust aligner for parallel texts based on a lexicon, wherein the remote-language pair and noise processing are considered. The alignment effectiveness of Champollion is high in noisy parallel texts, and some results on them reach about 97% precision. Up to now, Champollion can have been ported to other language pairs besides the Chinese-English language pair.

To attain multiple to multiple alignments, Champollion used about ten parameters, such as alignment penalties except 1-1 alignment, length penalty, and the weights in the similarity function. However, those parameters and weights are not optimal due to that they are decided empirically.

Thus, to improve the performance of Champollion by optimizing those parameters and weights has an important value in practice. Especially, in the research of SMT for patent texts, we found there are large parallel texts for patent, but the results of sentence alignment included some incorrect pairs, which introduced much noise to the next processing in SMT. Therefore, to construct SMT system with high quality, we require the precision of sentence alignment is high, but the recall of it can be a secondary factor. Thus, the performance of pre-processing can be enhanced effectively in SMT system by improving the precision of Champollion. Further, we can improve the total quality of SMT.

For the numerical optimizing, particle swarm optimization (PSO) is an important method in artificial intelligence, especially in the processing for non-convex function and multimodal function, which is superior to Downhill algorithms due to that it is not easy to fall into local optima, and it can find global optima and near global optima fast with low cost.

Our research towards optimizing Champollion improved the precision of it. Further, we can improve the pre-processing of SMT for patent texts. Thus, this paper presents a method to improve Champollion using PSO.

The rest of the paper is laid out as follows: Section 2 gives a brief overview of related work; in Section 3, first, we describe the Champollion algorithm and PSO algorithm; next, we present a method to optimize the Champollion algorithm; Section 4 provides the experiments and the results; and Section 5 concludes the paper.

2 Related Work

A cost-effective approach was presented for creating a large amount of parallel text resources [3], which recognized parallel document pairs from sources with potential parallel text, using Bilingual Internet Text Search (BITS) and Champollion alignment systems.

The word sense disambiguation is improved by using a constraint solver (MINION

tool) and rules formed by using CLIPS language on the aligned meaning of a word [4]. Those alignments are done by using Champollion.

For the English-Chinese bilingual corpora from Web mining, a reverse alignment and verification algorithm was proposed [5], which combined the word-level alignment and Champollion-based sentence-level alignment from word level to sentence level, indicating the more ideal effect.

To collect automatically high-quality parallel bilingual corpora from the web, Zhang et al. presented a method using multiple features to identify parallel texts [6]. The method used a *k*-nearest-neighbor classifier on those features, including the file length, the file structure, and the translation of the web page, which was calculated using the number of sentence alignment aligned by Champollion.

For sentence-level alignment, Peng et al. presented a Fast-Champollion algorithm [7]. The method utilized the advantages of both length-based and lexicon-based algorithm, and split the input bilingual texts into small fragments to accelerate the alignment.

3 Methods

3.1 Champollion Algorithm

It is impossible to check manually the parallel texts extracted from different format documents due to that the scale of them is so large. Thus, a robust method named Champollion was presented, which can detect the noise and recover from the mistake fast, whose characteristics are as follows:

1) Presume the data having noise, thus, the segments are multiple to multiple alignments.

2) Assign different weights to the translation words.

3) Recognize translation words using translation lexicon, and the statistic of translation words are utilized to identify corresponding sentences.

4) Assign large weight to the translation with small word frequency to calculate the similarity of two segments.

5) Penalize the alignment with a mismatch in sentence length.

6) Penalize the alignments other than 1-1 alignment.

7) Search optimal alignment segment with maximum similarity using a dynamic programming (DP) algorithm.

The idea of *tf-idf* (term frequency - inverse document frequency) weight used in Information Retrieval is borrowed by Champollion to calculate the similarity of segments, which can be expressed as follows.

$$\begin{aligned} & sim_1(E,C) = \sum_{i=1}^{k} \lg(stf(e_i^{'}, c_i^{'}) \cdot idtf(e_i^{'})) \\ & s.t. \quad idtf = T / \#tf\_doc \\ & \qquad stf = \#tf\_segm + 1. \end{aligned} \tag{1}$$

Where $E = \{e_1, e_2, ..., e_{m-1}, e_m\}$ and $C = \{c_1, c_2, ..., e_{n-1}, e_n\}$, and e_i and c_j are word tokens. There is a translation pair between two segments, e.g., $P = \{(e_1^{'}, c_1^{'}), (e_2^{'}, c_2^{'}) ... (e_k^{'}, c_k^{'})\}$. T is the number of total tokens in documents;

$\#tf\_doc$ is the number of tokens appeared in the document; $\#tf\_segm$ is the number of tokens occurred in the segment.

Champollion can handle the noise effectively by using the penalties, which can be expressed as follows.

$$sim(E,C) = sim_1(E,C) \cdot alig\_pena_{ij} \cdot leng\_pena(E,C). \quad (2)$$

Where $alig\_pena_{ij}$ are the parameters for the penalties. We can express it as follows further.

$$alig\_pena_{ij} = \begin{cases} 1 \text{ if } (i,j) \in \{1-1\} \\ p_{ij} \text{ if } (i,j) \notin \{1-1\}. \end{cases} \quad (3)$$

Where the p_{ij} is the penalties for the alignments other than 1-1 alignment, e.g., 1-2, 2-1, 1-3, 3-1, 1-4, 4-1, etc. Up to now, Champollion has been only used values empirically for these penalties. $leng\_pena(E,C)$ is the length penalty, which is the function of the length of segments, e.g., E is the segment of source language, and C is that of target language.

Champollion uses the DP algorithm to calculate the maximum similarity of two segments. Before the calculation, it tokenizes both sides of the parallel text, such as word segmentation for Chinese, and stemmer for English.

3.2 PSO Algorithm

Kennedy and Eberhart presented the standard PSO (SPSO) algorithm in 1995, which can simulate the behavior principle to construct an intelligence optimization algorithm for the numerical problem. In [8], a formal description can be concluded as follows.

Assuming the number of the parameters is D, we can assign the number of dimension of search space is D. Then, some D dimension vectors used in PSO are as follows.

1) The position parameter of particle is x.
2) The best position of x found in previous search is p.
3) The velocity parameter of particle is v.
4) The current best position of particle swarm is g.

The particle swam uses a topology structure to maintain the communication between the particles, some of which used commonly are as follows.

1) N-1 neighbor structure, where each particle can communicate with all other particles.
2) Ring structure, where each particle can communicate with left and right particles.
3) The neighbor structure initialized randomly.

The steps of standard PSO can be described as follows.

1) The velocity of particles varies due to the influence of the communication of particles as follows.

$$v = \otimes\alpha(v)\beta(p)\gamma(g). \quad (4)$$

Where α is an action on the vector v; β is an action on the vector p; γ is an

action on the global optimal vector g; finally, a combination operation $\otimes$ operates on these vectors to form a new velocity vector v.

2) Each particle flies to new position after the update of vector v as follows.

$$x = x \oplus v. \tag{5}$$

Wherein the $\oplus$ is a vector addition operation.

3) A clamp function can draw the particles flew out of the search space back into it as follows.

$$x = \eta(x). \tag{6}$$

Wherein η is a clamp function, which modifies the current position of the particle.

4) When the velocity of the particle is large, the granularity of the search will be large, resulting in some important areas ignored. Thus, a clamp function for this velocity is necessary as follows.

$$v = \delta(v). \tag{7}$$

Wherein, δ is another clamp function, which modifies the current velocity of the particle. In SPSO, these functions and actions can be specialized as follows.

$$\begin{cases} \alpha(v) = c_1 \cdot v \\ \beta(p) = c_2 \cdot (p - x) \\ \gamma(g) = c_3 \cdot (g - x). \end{cases} \tag{8}$$

Wherein the c_1, c_2, and c_3 are random number distributed uniformly, and the final position and velocity can be expressed as follows.

$$\begin{cases} v = \alpha(v) + \beta(p) + \gamma(g) \\ x = v + x. \end{cases} \tag{9}$$

3.3 Improve Champollion Using PSO

As the Subsection 3.1 described above, the penalties p_{ij} are multiple value decided empirically. On the other hand, the three parts on the left of Equation (2) have the same weights, which are also empirical. These empirical values are not global optima. Therefore, we can optimize them to improve the performance of the Champollion, i.e., the precision of sentence alignment.

The paper uses the SPSO algorithm to optimize these empirical parameters and weights, which called Champollion algorithm using those values, generated by SPSO to train on the training data set, and evaluated the results of sentence alignment. Then, it fed the evaluation values back to the SPSO as the fitness of the particles. After iteration, we obtained the optimal parameters and weights.

In fact, we hope the precision of the sentence alignment is maximum due to that the parallel texts are abundant in SMT for patent texts, and expect that the noise in results of sentence alignment is less. Therefore, less noise is introduced into the next steps in SMT. Thus the reliability of the sentence alignment will be improved even if the recall of it is influenced to some extent.

F1 is a popular evaluation measure, which can improve the generalization of the model on the training data set by integrating both precision and recall. Therefore we used F1 as the evaluation value on training. After training, we can use precision to judge the performance of sentence alignment on different testing data sets compared with Champollion method. The F1 can be expressed as follows.

$$\begin{aligned} precision &= tp/(tp+fp) \\ recall &= tp/(tp+fn) \\ F_1 &= \frac{2\cdot recall\cdot precision}{recall+precision} = \frac{2\cdot tp}{2\cdot tp+fn+fp}. \end{aligned} \tag{10}$$

Where:

(a) *tp* is the "true positive", which is the number of the positive instances;

(b) *fp* is the "false positive", which is equal to the number of the negative instances judged as positive instance;

(c) *fn* is the "false negative", which is equal to the number of the positive instances judged as negative instance.

As the Equation (3) describes, the $alig\_pena_{ij}$ optimized by the paper are as follows.

Table 1. Alignment Penalty Parameters in Champollion

No	Para Name	Para Meaning	Init value
1	penalty21	2-1 alignment penalty	0.95
2	penalty12	1-2 alignment penalty	0.95
3	penalty22	2-2 alignment penalty	0.85
4	penalty31	3-1 alignment penalty	0.94
5	penalty13	1-3 alignment penalty	0.94
6	penalty41	4-1 alignment penalty	0.92
7	penalty14	1-4 alignment penalty	0.92
8	penalty23	2-3 alignment penalty	0.92
9	penalty32	3-2 alignment penalty	0.92
10	penalty33	3-3 alignment penalty	0.92

On the other hand, to obtain optimal results, we rewrite the Equation (2) using three weights as follows.

$$sim(E,C) = r_1\cdot sim_1(E,C)\cdot r_{ij}\cdot alig\_pena_{ij}\cdot r_2\cdot leng\_pena(E,C). \tag{11}$$

Where the r_1, r_{ij}, r_2 are three weight values for $sim_1(E,C)$, $alig\_pena_{ij}$, and $leng\_pena(E,C)$ correspondingly, and $i,j\in\{1\sim 4\}$, optimized by SPSO. Thus, weights of *stf* and *idtf* wrapped by sim_1 can be optimized, and weight of length penalty is also optimized with the alignment penalties together.

4 Experiments

The experimental data sets included training, general test, and patent test data sets, whose characteristics concluded as follows.

Table 2. The Characteristics of Data Sets

No	Dataset	Usage	Cn Num	En Num

1	198706005	training	218	172
2	200110006	test	126	237
3	890621f	test	927	893
4	921008f	test	429	425
5	930422f	test	342	366
6	UN19930101_020	test	457	402
7	UN19990209_010	test	943	926
8	Patent text	test	1272	1775

Where: the sentences are ended with punctuations { '?', '!', '.', '\n' }.

From Table 1, we can see that the number of Chinese sentences are very different from the number of English sentences, where the data set from No. 1 to No. 7 are carried within the Champollion[1], including their gold standard alignment, which are freely available, and the No. 8 is the data set provided by China patent information center, including its gold standard alignment.

The configuration of SPSO is as follows.

Swarm Size: 5;
Dimension number: 12;
Inner loop: 10;
Outer loop: 5;
Minimum position: 0.1;
Maximum position: 1.

For the parameter c_1 in Equation (8), we used empirical value introduced in [9], which can be expressed as follows.

$$\begin{aligned} & c_1 = 2/(c_{exploit} - 2 + \sqrt{{c_{exploit}}^2 - 4 \cdot c_{exploit}}) \\ & st.\ c_{exploit} = 4.14. \end{aligned} \tag{12}$$

For the parameters c_2 and c_3, we used chaos number, which were generated differently each times. Thus, the SPOS can have a more powerful ability to search the problem space, which can be expressed as follows.

$$\begin{cases} c_2 = chaos(0, c_1 \cdot c_{exploit} / 2) \\ c_3 = chaos(0, c_1 \cdot c_{exploit} / 2). \end{cases} \tag{13}$$

Wherein the $chaos(\cdot)$ can be expressed as follows.

$$\begin{aligned} & chaos(a,b) = a + x_n \cdot (b-a) \\ & st.\ x_n = x_n \cdot m_u \cdot (1 - x_n) \\ & \quad x_0 = 0.67777;\ m_u = 3.99999 \end{aligned} \tag{14}$$

The $\eta(x)$ in Equation (6) clamped the position x between the minimum position and maximum position. In the same time, it resets the velocity of the x in that dimension out of the problem space to zero. The $\delta(v)$ in Equation (7) is ignored in the

[1] Champollion Tool Kit V1.2, http://sourceforge.net/projects/champollion/

experiments.

Before the optimization and after it on the training data set, the results are as follows.

Table 3. The Results Before and After the Optimization on Training Data Set

No	Algorithm	Precision	Recall	F1
1	Champollion	0.789216	0.838542	0.813131
2	PSO-Champollion	0.931818	0.854167	0.891304

From Table 3, we see that the precision, recall, and F1 values after optimization are higher than that before it, and the optimal parameters and weights are as follows.

Table 4. The Optimal Parameters and Weights

No	Name	Usage	Optimal	No	Name	Usage	Optimal
1	r1	*stf* and *idtf*	0.482933	7	r2	Length penalty	0.828838
2	r12	Alignment	0.205829	8	r13	Alignment	0.157388
3	r21	penalty	0.995646	9	r31	penalty	0.474738
4	r22		0.345076	10	r33		0.278365
5	r23		0.540765	11	r14		0.46898
6	r32		0.738044	12	r41		0.533589

To test the effectiveness of the proposed method, we directly applied the optimal parameters and weights to the Champollion on the test data sets, and the results are as follows.

Table 5. The Results on the Test Data Sets

Dataset No	Algorithm No	Precision	Recall	F1
2	1	*0.932143*	0.925532	*0.928826*
	2	0.916335	0.815603	0.863039
3	1	0.88821	0.879758	*0.883963*
	2	*0.919153*	0.826125	0.870159
4	1	0.914172	0.906931	0.910537
	2	*0.958606*	0.871287	*0.912863*
5	1	0.954869	0.954869	0.954869
	2	*0.97549*	0.945368	*0.960193*
6	1	0.911647	0.947808	0.929376
	2	*0.965885*	0.94572	*0.955696*
7	1	0.972718	0.971698	*0.972208*
	2	*0.985976*	0.958071	0.971823
8	1	0.797508	0.780854	*0.789093*
	2	*0.8238*	0.69275	0.752613

From Table 5, we see that the best F1 values using the two algorithms had a proportion 4:3, indicating an unchanging effect. However, the precisions of the two algorithms had a proportion 1:6, indicating the results were superiority after optimization, which reduced the noise generated by incorrect sentence alignments. Furthermore, this situation is especially suitable to the requirement of SMT for patent texts (See data set No. 8).

We guessed that the performance on the data set No. 2 after the optimization decline slightly is due to that the year between No. 2 and training data set is farther than

that between others and training data set. Therefore, the content of the data set was changed largely (See Table 1).

5 Conclusions

For the aligner Champollion in SMT for patent parallel texts, the paper presents an intelligent optimization method, which used SPSO to optimize the alignment penalties, weights, and length penalty used in Champollion. The method improves the precision of sentence alignment of Champollion effectively, and reduces alignment noise, which is conducive to the subsequent steps in SMT system, e.g., the model training and tuning.

The algorithm proposed in the paper can also be applied to the parameter optimization in other related fields, such as cross language information retrieval, and word disambiguation. For different types of parallel texts, how to use sampling algorithms for generating combinational data set as training data set, to improve the generalization performance of the algorithm on different types of parallel texts will be the future work.

Acknowledgments

This work was supported by the Hi-Tech Research and Development Program of China (2012AA011104), and Postdoctoral Science Foundation of China (Grant No.: 2013M540125, 2013M530026).

References

[1] Yamada Kenji and Kevin Knight. A Syntax-based Statistical Translation Model[C]// Proceedings of the 39th Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics. 2001.

[2] Xiao. Y. Ma. Champollion: a Robust Parallel Text Sentence Aligner[C]// In Proceedings of LREC-2006: Fifth International Conference on Language Resources and Evaluation. 2006: 489-492.

[3] Kazuaki Maeda, Xiaoyi Ma, and Stephanie Strassel. Creating Sentence Aligned Parallel Text Corpora from a Large Archive of Potential Parallel Text using BITS and Champollion[C]// In LREC.2008

[4] Romika Yadav and Rashmi Manker. Improvement of Word Sense Disambiguation Using MINION[J]. International Journal of Engineering, 2013, 2(3).

[5] Yue. J. Zhang, Lei Cen, Qing. X. Wu, Cheng Jin, and Tao Zhang. Fusion of Reverse Hierarchical Alignment for Web-based English-Chinese Bilingual Dictionary Construction[C]// In Machine Learning and Cybernetics (ICMLC), 2010 International Conference on IEEE. 2010. (3): 1288-1293.

[6] Ying Zhang, Ke Wu, Jianfeng Gao, and Phil Vines. Automatic Acquisition of Chinese-English Parallel Corpus from the Web[J]. In Advances in Information Retrieval, 2006, 420-431.

[7] Li Peng, Maosong Sun, and Ping Xue. Fast-Champollion: a Fast and Robust Sentence Alignment Algorithm[C]// Proceedings of the 23rd International Conference on Computational Linguistics: Posters. Association for Computational Linguistics. 2010.

[8] Maurice Clerc. Binary Particle Swarm Optimisers: Toolbox, Derivations, and Mathematical Insights[DB/OL]. [2005] http://hal.archives-ouvertes.fr/hal-00122809/en/

[9] Maurice Clerc. Stagnation Analysis in Particle Swarm Optimization or What Happens When Nothing Happens[R]. Technical report, http://hal.archives-ouvertes.fr/hal -00122031. 2006.

作者联系方式：熊文 北京市北太平庄路 25 号中国专利信息中心 100875 电话（18601930389） xiongwen@cnpat.com.cn