The Design of Web Based Machine Translation Server Based on Grid Infrastructure

Fai Wong, Kwok Kit Leung
Faculty of Science and Technology, University of Macau, Macao
{derekfw, ma56541}@umac.mo

Abstract

This paper presents the research work in designing the architecture of translation server of Internet based machine translation system by using the infrastructure of grid. Where the translation server is able to access a LAN (or WAN) whose spare computing resources could be employed to accomplish the massive translation works generated by overwhelming user requests. According to the characteristics of different analytical tasks in a translation system, the whole process can be factorized into a series of tasks, where some computational tasks can be decomposed into highly independent and making the processing highly parallelizable. Two different configurations of distributed tasks over the network environment have been constructed in our current study.

1. Introduction

The objective of machine translation (MT) system is to overcome the language barrier among the people for globalization of information available in different languages [1]. The other important goal is to provide effective tools for the professional translators to improve their daily work in a more efficient way. The development of stand alone or PC based (local based) machine translation systems has been the early objective to provide professionals a translation workbench through the use of computer technology. In the fast evolvement of information technology, language technology, and web technology, in particular network computing, the World Wide Web seems to be an ideal environment for machine translation [2]. However, different from the monolithic local machine translation systems, the network scenario with embedded machine translation functionality is the vision which might turn MT as well as machine aided translation to challenging new application domains such as interlingua-based email transmission, instant messaging translation, interactive machine translation, and the translation of Web pages or documents on demand [3]. This is foreseeable the workloads of this kind of translation system will be very heavy once the service is published and opened to every Web user.

Based on these challenging considerations, in this paper, we present a Web based machine translation system that takes advantage of network technology and existing local based machine translation systems [4][5] with different characteristics to solve the problem of massively parallel natural language processing. From technical point of view, the implementation of network-based MT system comprises two parts: rendering the front end in browser which provides service interface to user, and establishing the back end translation server. However, the main concern of this study will focus in the design of translation server which allows to access to a LAN whose spare computing resources could be employed to accomplish the massive translation loads by controlled degradation of performance in case of overwhelming user requests. By reviewing the nature of MT system, the translation of documents composes of a series of processing tasks from different analysis levels of context. Depending on characteristic and dependencies of different analysis processes, some computational tasks required to perform the processing that can be decomposed into highly independent subtasks making this processing highly parallelizable. Such problems provide the luxury of being able to choose between different configurations of distributed tasks over the network environment, especially the architecture based on the topology of grid. Furthermore, this involves the advances in parallel and distributed processing. With the further study of these computing technologies and natural language characteristics, we will significantly advance the research on massively parallel natural language processing.

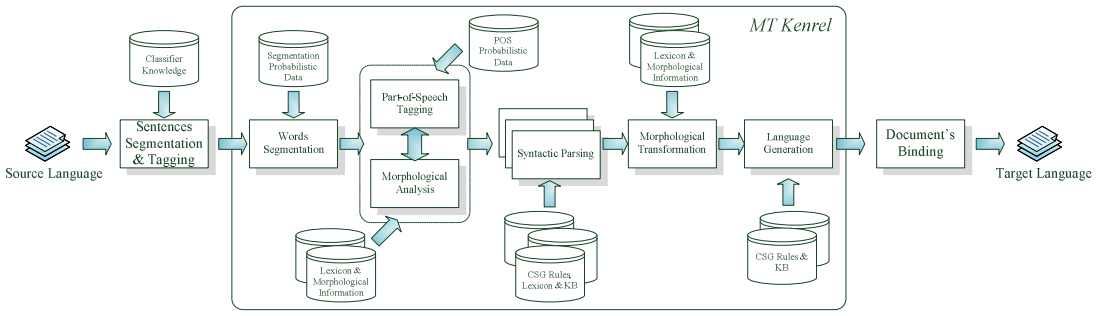

Figure 1. The flow of different language processing tasks in machine translation system

This paper is organized as follows. Section 2 discusses the parallelizability of different language processing tasks in MT system that we adapted for this research purpose. Section 3 discusses the characteristic of the infrastructures between cluster and grid. Two configurations of translation server are evaluated in section 4. Section 5 concludes the study, and directions for further research are given to end this paper.

2. Parallelizability analysis in machine translation

When analyzing a problem for discovery of potential parallelizability, there are a few standard characteristics that can measure which give some idea of parallel computational potential. In our case, two such characteristics are examined: 1) one of the most useful characteristic to analyze is a process's runtime I/O profile. If a process spends the majority of its time reading and/or writing files then there is no benefit to splitting computation among multiple processors; and 2) the granularity of tasks, if it has been determined that the majority of a process's runtime is spent on computation as opposed to I/O, it is then necessary to determine the possibilities for task decomposition.

In response to the first property for the MT tasks, the time spent in I/O of the tasks is limited and is only happened at processes' initialization. Regarding to the second property of granularity of tasks, this measure actually relies on the design of analyzed system. In machine translation, different translation approaches may lead to different design architecture due to type of data knowledge used to facilitate the language translation. For example, in Rule-Based MT system [6], it translates sentences by doing deep analysis through a series of linguistic analysis and structural transformations from one language to another as the target translation, based on a set of linguistic rules, where rules are translated in a linguistic way. In Statistic-Based MT [7], the translation is determined by estimating the probabilities between the translation of words and the ordering of the sentences based on parallel corpora. This approach usually bases on a single (or tightly coupled) mathematical model(s) to achieve the objective of sentence translation. For Example-Based MT approach [8], it adapts the analogical translation examples stored in the parallel corpora to reproduce the translation for the input sentences according to the similarities of sentential constituents. Thus, the kernel processes of Example-Based MT comprise the modules of examples *searching* and corresponding translations *adaptation*.

In our case, the MT system used for the study is based on Rule-Based approach. Different from the conventional transfer architecture [1], we applied the Constraint Synchronous Grammar (CSG) [5], as the kernel of the MT system for modeling the structural relationships between multiple languages in parallel. That is, during the translation process, the recognition of syntactic structure of source language and the transformation to the target language structure can be accomplished at the same time using the same set of CSG productions.

Reviewing the translation components of the MT system as depicted in Figure 1, besides the modules of Sentences Segmentation & Tagging and Document Binding, to identifying the boundaries of sentences for feeding into the (sentence-based) translation engine and assembling the target document in the correct format, the translation kernel consists of six major components: Word Segmentation, Morphological Analysis, Part-Of-Speech (POS) Disambiguation, Syntax Parsing and Transformation, Morphological and Language Generation.

2.1. Word segmentation

Unlike Western languages, Chinese is written in a continuous way without any delimiter to explicitly specify the word boundaries. Therefore, for any Chinese Information Processing (CIP) system like machine translation, web information retrieval, etc., the first and most essential step is word segmentation to

find out the boundaries of words within sentences. The unknown words and ambiguities have been the main obstacles to the make the word segmentation difficult. In order to complement the problem in our real application, our strategy is to pick the *N*-best rough segmentations, aimed to obtain a high recall rate of including the correct segmentation. Then let the upcoming analysis processes determine and choose the correct sentence based on other analytical information. To tackle this problem, probabilistic *N*-shortest-paths model [9] has been applied to do the segmentation.

For a given Chinese sentence, a directed graph is constructed with each of its atomic characters as the vertices ($V_1, V_2, ..., V_n$). Edges between the vertices are determined by probabilities of the atomic characters or the combinations of the words obtained in the Chinese corpus. Let W be one of the possible results of the segmentation for the Chinese sentence C, then the probability of W, given C is defined as:

$$P(W \mid C) = \frac{P(W)P(C \mid W)}{P(C)} \quad (1)$$

Since the probability of the Chinese sentence $P(C)$ is a constant, and the probability of C, given W must be 1, the objective is to determine the N different segmentations which have the N largest probabilities of $P(W)$. Suppose that a possible segmentation sequence W consists of $w_1, w_2, ..., w_m$ words, then the probability $P(w_i)$ can be approximated as:

$$P(w_i) \approx \frac{(k_i + 1)}{(\sum_{j=0}^{m} k_j + V)} \quad (2)$$

k_i is the number of occurrences of w_i and V is the number of word types in the training corpus. Smoothing is applied by adding a constant in the numerator by taking into consideration that w_i may not appear in the training corpus. By assuming that the context within the sentence is not considered for simplicity, the best word sequence W can be computed as:

$$\arg\max_{w} P(W) = \arg\max \prod_{i=1}^{m} P(w_i) \quad (3)$$

$$\approx \arg\max(\prod_{i=1}^{m} \frac{k_i + 1}{\sum_{j=0}^{m} k_j + V}) \quad (4)$$

Based on the model, the possible segmentations that have N largest probabilities of $P(W)$ are selected as the candidates to be passed to the subsequent processes. Considering the case of parallelism, multiple instances of this process can be run in parallel to massively segment multiple sentences at the same time.

2.2. Part-of-speech disambiguation

In natural language processing such as machine translation and language understanding, knowing the *part-of-speech* (POS) of words is an important step in discovering the linguistic structure of sentences. This information facilitates the higher-level analysis, such as recognition of noun phrases and other syntactic patterns in text. It is an essential task in machine translation that provides the necessary information to further analyze the syntax of sentence. This seems to be a simple problem, but it is actually a very difficult problem. It suffers from serious problems of part-of-speech ambiguities and unknown words due to variations of morphology. In our system, the tagger is constructed based on probabilistic model with an extension to interpolate the orthographic features of words for predicting the correct POS [10].

That is, let T be the most likely POS sequence of a given particular word sequence, W. The probability of each sequence can be approximated as a product of the conditional probabilities of each word or tag given all of the previous tags:

$$P(T \mid W) \approx \prod_{i=1}^{n} P(t_i \mid t_{i-1}, t_{i-2}, ..., t_1) P(w_i \mid t_i, ..., t_1 w_{i-1}, ...w_1) \quad (5)$$

Typically, two assumptions can be made to cut down the number of probabilities to be estimated. One assumes word w_i depends only on tag t_i, and tag t_i depends only on previous tag t_{i-1}, since local context is sufficient. Then using a bi-gram model, we have:

$$\arg\max_{T} P(T \mid W) \approx \arg\max \prod_{i} P(t_i \mid t_{i-1}) P(w_i \mid t_i) \quad (6)$$

To resolve the unknown words problem, the orthographic features of words, mainly the lexical *suffixes* (with length of 3 to 5 characters) and prefix *capitalization*, i.e. $f_i \in (cap_i, suf_i^3, suf_i^4, suf_i^5)$, will be interpolated into the word probability, defined as:

$$P_{interp}(w_i \mid t_i) \approx \lambda_1 P(w_i \mid t_i) + \frac{P(w_i)}{P(t_i)}[\lambda_2 P(t_i \mid cap_i) + \lambda_3 P(t_i \mid suf_i^3) + \lambda_4 P(t_i \mid suf_i^4) + \lambda_5 P(t_i \mid suf_i^5)] \quad (7)$$

Where $\lambda_i (i = 1..5)$ are the interpolation coefficients. From the design point of view, a parallel approach for this process can be deployed to determine the part-of-speech of words for different portions of an input document at the same time to accelerate the overall tagging performance.

2.3. Morphological analysis

The function of morphological analysis in machine translation is to recover the canonical form of lexical item for Western languages such as English and Portuguese, as well as to analyze the associated linguistic information based on word morphemes. Actually, the analysis of compositional word is considered as syntactic analysis, since the inflectional morphology defining the possible variations on the root of word reflects the grammatical meaning and semantic information, such as gender, person, tense, mood, number, and grammatical category, etc. Besides the objective to conclude the conventions of morpheme variations hence to reduce the size of lexical dictionary from keeping the inflectional paradigms in full, another important intention is to resolve the unknown word problem by using the software analysis approach to dynamically recover the canonical form of lexicon and extract the embedded grammatical information.

In case of parallelism, the processes of part-of-speech tagging and morphological analysis are independent, and can be run in arbitrary order, as depicted in Figure 1. Similarly, the morphological analyzer can be run in multiple instances in parallel to massively analyze the word morphemes for documents.

2.4. Syntactic parsing and transformation

In machine translation, to analyze the structure deviations of languages pair hence to carry out the transformation from one language into another as the target translation is the kernel, and this requires a large amount of structural transformations in both grammatical and concept level. The problems of syntactic complexity and word sense ambiguity have been the major obstacles to produce promising quality of translation. As stated, we applied the Constraint Synchronous Grammar (CSG) [5] to model the syntax relationship between multiple languages in parallel based on the formalism of Context Free Grammar (CFG) to the case of synchronous. In bilingual case, the CSG formalism consists of a set of production rules that describes the sentential patterns of the source text and target translation patterns in the form of:

S → NP_1 PP NP_2 VP* NP_3
{ [NP_1 VP *a* NP_3 NP_2] ; VP_{cat} = *vb1* &
PP = "把" & $VP_{s:sem}$ = NP_{1sem} &
$VP_{o:sem}$ = NP_{2sem} & $VP_{io:sem}$ = NP_{3sem}
[NP_1 VP NP_2 *em* NP_3] ; …
}

In this production rule, it has two generative rules associated with the sentential pattern of the source *NP_1 PP NP_2 VP NP_3*. The determination of the suitable generative rule is based on the control conditions defined by rule. The one satisfying all the conditions determines the relationship between the source and target sentential pattern. The asterisk "*" indicates the head element, and its usage is to propagate all the related features/linguistic information of the head symbol to the reduced non-terminal symbol in the left hand side. Their relationship is established by the given subscripts and the sequence is based on the target sentential pattern. In this model, semantic information is represented by feature descriptors (FD) which give additional flexibility in defining CSG rules for establishing agreements in syntactic and sub-categorization dependencies. Feature unification is performed during the parsing stage. If FDs of each lexicon word or lexical are compatible with each other, i.e. there are no conflicts on the value of all the attributes defined, unification succeeds. Then the production rule is applied and a new FD is constructed for the reduced symbol as a valid syntactic constituent. From the translation point of view, the syntactic structure of target sentence can be determined at the same time once the source sentence is successfully parsed. This process is very time consuming because of the ambiguities of language. Thus, the parallelism of this task can greatly improve the performance of system to the translations in large scale.

2.5. Generation of the translation

The generation process takes the responsibility to render the translation of the input sentence by referencing the set of generative target sentential patterns that were selected previously. Moreover, in order to ensure that the system generates perfectly the translation in target language grammatically, we employ unification of Functional Descriptors (FD) as a validation operation for each node, which was

constructed for each constituent node in the parsing stage. This includes the change of word morphology based on the set of grammatical agreements such as number, gender, tense, and categories of person, and the render of target sentence based on syntactic and semantic constraints, as examples shown in Figure 2.

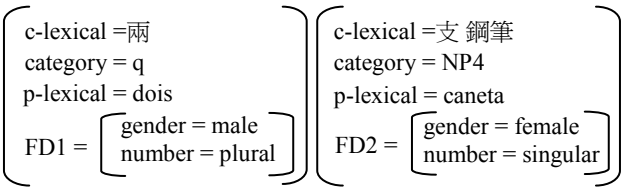

Figure 2. AVMs of words "*兩*" and "*支 鋼筆*"

Unification of FD1 and FD2 will fail as the gender and the number are different. In such a case, necessary conversions are performed so that FD1 and FD2 will be compatible with each other. Therefore, the generated result for "*兩支鋼筆*" is "*duas canetas*" (two pens).

This task can be parallelized in the generation of sentences for a document. According to the parallelism analysis of the translation tasks, we found that the major components can be easily reconfigured into parallelizable computation without putting too much effort on it.

3. Cluster versus grid

The first step in designing translation server over the network environment comes to the choice of the computing infrastructure: *cluster* or *grid*. Although cluster and grid share the common property to solve problem by using two or more computers through the technologies of parallel and distributed computing, they have different computational nature and configuration requirements [11]. From the development point of view, grid gives a larger degree of flexibility to use spare computing power which is loosely connected on network. The computers in cluster are normally contained in a single location. Most importantly, the computers that are part of a grid can run different OS and have different hardware whereas the cluster computers are required to have the same hardware and OS. It is clearly a trend that as workstations and network infrastructure grow increasingly less expensive, more workstations will be available on a LAN than can be interactively used by personnel. The grid model of computation intends to take the advantage of these unused resources.

3.1. Translation grid framework

There has been a great deal of recent interest in grid and peer-to-peer computing [11]. Unlike cluster computing, there is no clear standard or even agreed upon standard model. Most researches focus on the generic case that a single global gird works on particular problem. There are many different possibilities for implementing grid based solutions according to the capabilities and possibilities presented by each situation. For our case, we implemented our own grid infrastructure based on web services, since it is the most ubiquitous. This allows us to communicate through the firewalls, and the access to resources available in more than one location can be easily achieved and implemented through HTTP. Therefore, the main grid component is a web ASP (or JSP) based program that connects via socket to an internal host running a job scheduler and feeds these jobs to computation nodes via HTTP request on the client side. Clients periodically poll the server for jobs, so it is a pull model of computation rather than a push model where servers initiate jobs on the clients.

4. Configurations of framework

In this section we carry out two experimental configurations for the translation framework to observe the effect of system performance in terms of the respond time of different translation tasks. Table 1 gives the runtime of different tasks in translating 1000 sentences from textbooks by using our original (Portuguese-Chinese) MT system. We found that most of the computations fall into the task of syntax parsing and transformation, around 83% of the translation time is spent in analyzing the syntax of source sentences and inferring the corresponding syntax in target language. The response time to translate all sentences is around 133 seconds (2m13s).

Table 1. The processing time of different tasks

Tasks	Times	%
Word Segmentation	4.3s	3.16%
Morphological Analyzer	4.4s	3.29%
POS Disambiguation	9.4s	6.91%
Syntax Parser & Transformer	1m53s	83.46%
Translation Generation	4.3s	3.18%

For experimental purpose, we allow the main grid component to use 5 PCs from LAN as the client nodes. The job scheduler is configurable and allows us flexibly control the behavior of nodes. For example, in our first experiment, according to nature of different

tasks and their processing time, we define four of the nodes to be the parsing nodes dedicated for syntax analyzing, while the other node is appointed to handle the rest of the processing. Under such configuration, we re-run the translations with the same set of 1000 sentences. The overall response time of the translation is around 27 seconds. The average processing time for parsing and other translation tasks are 22.7 seconds and 6.8 seconds respectively.

In the second experiment, considering that most of the processing time is taken by the syntax analyzer, and the runtime of other tasks is very small, all nodes are configured as the translation nodes. Where each node is able to perform the full translation once a sentence is received from the server. This is similar to that we have multiple translation engines (systems) running parallel behind the server by using the spare resources from a LAN to accomplish the massive translation jobs. Based on this configuration, the same input sentences are used to test the framework. The response time of the translation server is 21.3 seconds, and the average processing time of each node is 19.89 seconds. The result is a little bit better than the previous one.

5. Conclusion

This paper presents the work in designing a Web based MT server based on grid infrastructure. From the parallelism point of view, possible processing tasks in MT system are analyzed and re-developed in parallel through the use of multi-threading technique to maximum the throughput of each subtask. Two configurations are constructed to compare and observe the performance of the translation framework. Currently, the translation server is based on a single Rule-Based MT engine, and this can be further extended to include multiple translation engines into the framework, such as the Example-Based MT and Translation Memory to further improve the translation performance.

6. Acknowledgements

The research work reported in this paper was partially supported by Science and Development Fund of Macau SAR under grant 041/2005/A and Center of Scientific and Technological Research (University of Macau) under Cativo: 5571.

7. References

[1] Hutchins, W.J., and H.L. Somers, *An Introduction to Machine* Translation, Academic Press, London, 1992.

[2] C.C. Hao, K.K. Leung, H.W. Tou, F. Wong, and Y.P. Li, "Knowledge Sharing in Network Based Portuguese-Chinese Translation System", *Proceedings of Symposium on Applied Science and Technology in Macau*, University of Macau, Macau 2004, pp. 43-51.

[3] J. Schütz, "Network-Based Machine Translation Services", *Proceedings of EAMT Machine Translation Workshop, TKE'96*, Vienna, Austria, 29-30 August 1996, pp. 147-159.

[4] F. Wong, C.W. Tang, M.C. Dong, Y.H. Mao, and Y.P. Li, "Example-Based Machine Translation Based on Translation Corresponding Tree Representation", *Proceedings of First International Joint Conference on Natural Language Processing (IJCNLP-04)*, Hainan Island, China, 2004, pp. 31-38.

[5] F. Wong, M.C. Dong, and D.C. Hu, "Machine Translation by Parsing Constraint-Based Synchronous Grammar", *Tsinghua Science and Technology*, Vol. 11, No. 3, 2006, pp. 295-306.

[6] W.S. Bennett, and J. Slocum, "The LRC Machine Translation System", *Computational Linguistics*, Vol. 11, No. 2-3, 1985, pp. 111-121.

[7] P.F. Brown, S.A.D. Pietra, V.J.D. Pietra, and R. Mercer, "The Mathematics of Statistical Machine Translation: Parameter Estimation", *Computational Linguistics*, Vol. 19, No. 2, 1993, pp. 263-311.

[8] R.D. Brown, "Example-Based Machine Translation in the Pangloss System", *Proceedings of 16th International Conference on Computational Linguistics (COLING-96)*, Copenhagen, Denmark, 1996, pp. 169-174.

[9] K.S. Leong, F. Wong, C.W. Tang, and M.C. Dong, "CSAT: A Chinese Segmentation and Tagging Module Based on the Interpolated Probabilistic Model", *Proceedings of The Tenth International Conference on Enhancement and Promotion of Computational Methods in Engineering and Science (EPMESC-X)*, Sanya Hainan, China, 21-23 August, 2006, pp. 1092-1098.

[10] F. Wong, S. Chao, M.C. Dong, and Y.H. Mao, "Interpolated Probabilistic Tagging Model Optimized with Genetic Algorithm", *Proceedings of Third International Conference on Machine Learning and Cybernetics*, Shanghai, China, 26-29 August, 2004, pp. 2569-2574.

[11] C. Ernemann, V. Hamscher, R. Yahyapour, and A. Streit, "On Effects of Machine Configurations on Parallel Job Scheduling in Computational Grids", *Proceedings of International Conference on Architecture of Computing Systems, ARCS*, Karlsruhe, Germany, 8-12 April, 2002, pp. 169-179.