A Customizable, Self-Learnable Parameterized MT System: The Next Generation

Keh-Yih Su and Jing-Shin Chang

Behavior Design Corporation
2F, No 5, Industrial East Road IV
Science-Based Industry Park
Hsinchu, Taiwan, ROC.

Email: {kysu,shin}@bdc.com.tw

Abstract

In this paper, the major problems of the current machine translation systems are first outlined. A new direction, highlighting the system capability to be customizable and self-learnable, is then proposed for attacking the described problems, which are mainly resulted from the very complicated characteristics of natural languages. The proposed solution adopts an unsupervised two-way training mechanism and a parameterized architecture to acquire the required statistical knowledge, such that the system can be easily adapted to different domains and various preferences of individual users.

1 Current Status

Most MT systems, currently, adopt a general-purpose kernel for every application. When different domains are encountered, they only swap the corresponding technical compound dictionaries, without tuning the system with the associated domain knowledge. Due to lacking the capability to economically acquire the huge and fine-grained knowledge[1] (both linguistic and real-world) required for different domains, the quality of those general purpose machine translation systems does not show much improvement during the last 50 years. As a result, the MT output quality usually cannot match user's expectations, and the threshold for user acceptance has not yet been reached. Since the usefulness of an MT system will be appreciated only when its quality exceeds the threshold, machine translation systems are still not widely used as a useful translation tool by many people.

In addition, various users and companies usually have different preferences in technical term translation, and show different tastes on translation output styles. However, most current machine translation systems still cannot adapt to the individual's preference. Lacking such adaptation capability will render the MT system making repetitive errors and thus inducing user frustration, because they have no way to teach the dumb system to stop generating those stupid errors. In contrast, many applications in other fields (e.g., speech recognition and on-line character recognition) are usually user adaptable. You can let the computer learn what you prefer and significantly improve the performance.

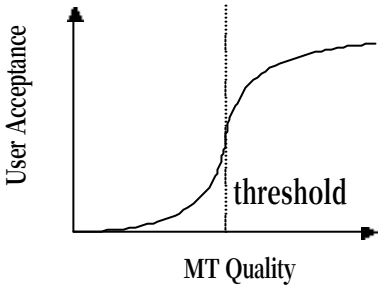

Figure 1: User Acceptance level *vs.* MT Quality

The direct consequence of failing to generate user acceptable result has two folds: the profit is marginal, and the market share is small. The low profit margin can be explained by the user acceptance level curve (Figure 1). It is observed that the user acceptance level curve usually appears as an S-shape. Therefore, there is almost no difference to the users when two systems are both below the threshold; and they will not give much credit to the improvement before the threshold is reached. Consequently, almost all MT systems are classified into the same category, from the users' point of view, even after having struggled for so many years. As a result, even more advanced machine translation systems, after having invested heavily in R&D expense, are unable to post reasonable price tags. Because they must be competitive in price with respect to those low end commercial MT packages, which are almost free of

[1] The required knowledge is mainly used to attack the main problems in natural language processing, i.e. ambiguity and ill-formedness.

Keh-Yih Su and Jing-Shin Chang, "A Customizable, Self-Learnable Parameterized MT System: The Next Generation," (invited paper) Proceedings of the Machine Translation Summit VII '99, pp. 182-188, Natl. Univ. of Singapore, Singapore, Sept. 13-17, 1999.

charge (you can buy a so-called MT system for less than four U.S. dollars in the trade show sometimes), when they are judged as no much difference by the users.

On the other hand, the low translation quality also makes a MT system harder to enlarge the scale of the business (i.e., to have more market share in the translation service sector), because considerable post editing will be required to amend the raw output. The post-editing demand, in turn, requires a lot of man power. However, the large human resource demand would impose a serious challenge in recruiting, training, and managing those post-editors; besides, how to balance the seasonal variation in translation demand will also be a big headache when many people are involved. Thus, the current MT systems do not provide enough competitive edge, hopefully obtained from the mass production, over those small human-translation agents who enjoy the advantages of lower operating cost and overhead.

2 Why Unsatisfied Quality?

There are several reasons why we still cannot get satisfied translation quality so far. The most frequently mentioned one is that the system does not possess enough required knowledge[2]. Although this is the underlying reason that many other factors should be attributed to, it is too broad to explain the phenomenon observed from the operating point of view. Therefore, we will take different views in this paper as follows.

Currently, most MT systems adopt a general-purpose kernel and then use it for every application. When different domains are encountered, they just adopt various corresponding technical compound dictionaries, without tuning the systems with associated domain knowledge; and it is hoped that the change of the domain lexicon will resolve the variation between different domains. Unfortunately, this simple approach usually does not work well as expected, since ambiguity resolution and ill-formedess handling also requires domain knowledge in many cases. Besides, the requirements for handling stylistic and grammatical variants are also different for various domains. Thus, the quality usually significantly drops when switching to a new domain which had not been well tuned for. The main reason why most MT systems adopt this simple approach is that the cost is usually quite high for tuning a conventional general purpose system into a specific domain. To do such adaptation, many rules and fine-grained knowledge must be modified, and a domain-identification front-end is also required when several sets of domain knowledge are available.

[2] This issue is very difficult to handle (both technically and economically) with the conventional rule-based approach, unless different paradigms such as Corpus-Based Statistics-Oriented (CBSO), to be described later, are adopted.

It seems that the problem can be avoided if only one domain is to be served by the MT system. In this case, we can just design a special purpose system such as the Canadian TAUM-METEO weather forecast system [Hutchins 86] to get high quality. However, from the market points of view, an MT system must have a large customer-base and serve for many projects at the same time in order to be economical and to survive. Therefore, it is almost inevitable for an operating MT system to include the projects from various domains. Under the constrain of not being able to economically tune the system knowledge, the developer is thus forced to use a general-purpose system with different technical dictionaries to serve all customers. Thus, as explained above, an unsatisfied system is almost inevitable under such circumstances.

In addition, conventional one-way training approach also contributes to the unsatisfied result. In conventional transfer-based MT systems, the transfer and generation knowledge or rules are strongly dependent on the source language. Such a one-way training strategy therefore often produces target sentences that are too literal and not nature to the native speakers; thus, it frequently results in low user acceptance ([Su 95]). Furthermore, the low satisfaction also resulted from the fact that most MT systems lack a systematic customization capability for customizing the system according to the user preference. Consequently, even an MT system can generate a readable output, it is still far from the desired result. Last of all, as many conventional MT systems do not possess the feedback mechanism to interact with the post editor, they not only produce undesired result at the first time, but also make the same stupid error repeatedly. Hence, a heavy post-editing is usually required in many practical applications.

From the above observations, it is clear that what the MT users actually need is a system that can really save translation time. In other words, they want to use it as a productive tool, instead of a toy for fun or for curiosity. This dream can only be achieved by the system that can produce the output that is very close to the final desired form. Therefore, the users not only want the system to be able to produce the good translation quality for various domains initially, they also want the system to be customizable, which can adapt itself to produce the preferred terms and style as wished. Furthermore, the system should be also self-learnable, that can learn a user's feedback and produce the output that approaches to the final desired form closer and closer as time goes by.

3 Strategies for Improving Quality

There are two possible directions to improve the quality of an MT system. The first strategy is keeping the problem to be the same, i.e., designing a general purpose high quality MT system as posted in the beginning of MT history, and trying to find the new and more powerful way to reach the goal, hopefully. Another strategy is to

admit that achieving two difficult goals (i.e., wide-coverage and high-quality) at the same time is not an easy task, if not infeasible, and then adopt the well-known divide-and-conquer approach to attack only one goal at a time. With this approach, we will not have one general purpose system; instead, we will have many special purpose systems and we will concentrate on only one specific domain at a time. That is, we decompose the original task into many subtasks, and thus have a simplified problem to be solved for each subtask; therefore, we lower the level of the hurdle. We will examine these two strategies in more details as follows.

As described above, the first strategy still adopts a general purpose system, and enhances its capability by including more and more real-world knowledge and theories, then hoping that the unsatisfied parts could be covered and resolved by these kinds of enhancement. Due to the difficulty encountered in acquiring and managing the knowledge consistently and economically, and integrating different knowledge sources systematically, unfortunately, this strategy did not show much success over the past 50 years. The enhancement for a particular problem usually introduces other unexpected problems, resulting in an unstable system performance, which is called the seesaw phenomenon during the system tuning process. As a result, the gain usually does not justify the cost required for enhancement. Even worse, it may end up with an unmanageably complicated system, which is very hard to maintain (even by the original developers).

As described before, another strategy is to simplify the translation tasks by concentrating on only one specific domain at a time (i.e., adopting the divide-and-conquer strategy). This approach is based on the observation that it is quite difficult to develop a wide-coverage high-quality system with current NLP technologies. Past experiences have shown that high performance MT is only achievable in restricted domains. One famous example is the Canadian TAUM-METEO weather forecast system [Hutchins 86]. If we can design various MT systems for each domain, then the threshold for user acceptance will be reached easier, which, in turn, will raise the profit margin and enlarge the market share of the MT services.

However, there is still one unsolved problem for adopting the second strategy. The knowledge acquisition cost for designing many special purpose MT systems is prohibitively high, as most MT systems are not parameterized. Fortunately, during the last decade, corpus-based statistics-oriented approaches [Su 96] have shown their power in automatic knowledge acquisition. The success of such approaches makes the task of knowledge acquisition much easier and cheaper, since heavy human involvement during the process can be avoided. Such corpus-based statistics-oriented approaches also provide us with a good chance to develop a highly parameterized MT system which is self-learnable and user-customizable [Su 96].

Such a paradigm shift, from the rule-based approach to the corpus-based statistics-oriented (CBSO) approach (which has been observed in various aspects of the NLP community during the last decade), is mainly driven by the following environmental impacts. First, people now have more and more online corpora available than ever. The CBSO approach represents and embeds the knowledge as many implicit probabilistic parameters, and automatically acquires the knowledge (i.e., the probabilistic parameters) from the corpus under a given statistical language model (i.e., probabilistic form). Therefore, the corpus is the main knowledge source; thus, the knowledge coverage rate will heavily depend on the corpus size for this kind of data-driven approach. As the corpus size has grown several orders of magnitude bigger (and still keep growing) in this Internet age, the induced knowledge is thus able to cover most language usage phenomena. People therefore no longer solely rely on linguists to develop and generalize the theory, originating from a small amount of text examples, in order to cover those unseen phenomena. Furthermore, the quick advance in computer technologies also makes the mass computation feasible, and those unsupervised training methods better suited.

The proposed highly ‘ parameterized’ MT system, implied by the CBSO approach above mentioned, not only reduces the system cost by acquiring the required knowledge automatically through unsupervised learning methods, but also makes the system to be customizable and self-learnable. Because when we want to produce an MT system for a special domain, we only need to re-estimate the required parameters from the related corpus from that domain. In the mean time, the capability to be self-learnable can be achieved by adjusting those parameter values through the user feedback. Therefore, such a system can easily provide us with the capability for adapting to user preferences in terminology and style in different domains according to the user feedback. This approach is also inspired by the success of other research communities (such as speech recognition), in which unsupervised training, domain adaptation and user adaptation are well studied.

Last, to produce natural and publishable translation output, the automatic acquisition process must be trained in an appropriate way so that the acquired translation knowledge is independent of the source language. Traditional knowledge acquisition methods are obviously inappropriate for this purpose, since the target generation knowledge often depends on the source-to-target transfer knowledge, which in turn depends on the result from source analyses. A two-way training method, which directly learns the best transfer mapping from the source construct and the final desired text, is therefore necessary. In the following sections, a new architecture, inspired by the above-mentioned philosophy, is proposed for the next generation MT in order to meet all the requirements simultaneously.

4 Architecture for a Self-learnable MT

To provide an MT system which is able to response to the requirements above mentioned, it should at least possesses the following capabilities:

High Quality: The system must be able to generate the output that closely approaches the final desired form.

Customizable: The system must be able to be adapted to different domains and various user-specific preference.

Self-learnable: The system must be able to include the user input as a kind of feedback knowledge, and re-train the specific user parameter set to fit the user better and better.

Low Development Cost: The system must keep the development cost low; otherwise, we can never get the investment back.

It seems that the architecture that can meet the above requirements simultaneously must possess the following characteristics:

Automatic Learning: The learning process must be able to learn the knowledge from large un-annotated corpora in order to reduce the knowledge acquisition costs.

Parameterizing: The system must be made easily changeable for various domains and various user's preference with a set of parameters.

Two-Way Training: The training process must be able to acquire source-independent translation knowledge, *via* unsupervised learning from the bilingual corpus, so that the translation will not be affected by the source language, and the system could get high-quality target translation.

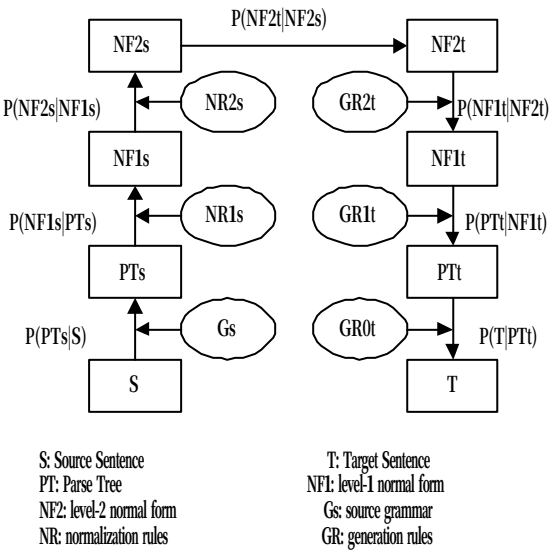

Figure 2: Architecture of a Parameterized MT

Figure 2 shows a system architecture that well fits the various concerns described above, which can be used to develop the next generation MT. In this figure, the source sentence is analyzed and the target sentence is generated phase-by-phase, each phase is characterized by an intermediate representation called a normal form of its previous phase. For instance, the source sentence is parsed into a parse tree, which is then normalized through a syntactic normalization process into a syntactically normalized form (denoted as the NF1 structure, level-1 normal form); the NF1 tree is then semantically normalized into a semantically normalized form, denoting as the NF2 structure (level-2 normal form) in the figure.

The ambiguity (including user preference), if any, is resolved using sets of (probabilistic) parameters. Such parameters might be the conditional probabilities used in parts of speech tagging or syntax disambiguation, and so on. Figure 2 shows that, for example, a system might use the conditional probability of a possible source parse tree (PTs), conditioned on a given input source sentence (S), as a parameter for disambiguation. Note that, all possible 'forms' of the possible analyses in the various phases are still expressed explicitly in terms of conventional linguistics representations, such as source grammar (Gs). This makes it easy to include well-developed linguistics formalism into the system, so that the developer can easily utilize such explicit knowledge. The disambiguation knowledge (including those for tailoring to special user preferences), on the other hand, is expressed, implicitly, in terms of the large set of above-mentioned parameters. This makes it easy to automatically learn the disambiguation knowledge in the general domain, and the user preference in a specific domain, with very little human intervention.

Such an architecture thus suggests a good way of corporation between traditional linguistics formalisms and the corpus-based statistics-oriented approaches in constructing a parameterized MT system. Because of those good characteristics, the requirements in parameterization, customization, and automatic learning can be satisfied easily with the above architecture. Besides, the requirement for self-learning can also be satisfied easily by adding a feedback loop to the system in the process of training the system parameters, as shown in Figure 3, where the discrepancy used for adjusting the system parameters could be measured in terms of post-editing costs to insert, delete or substitute some target lexicon ([Su 92]). The feedback loop could also be used to learn the preferred lexicon or syntactic styles of a particular user preference.

Keh-Yih Su and Jing-Shin Chang, "A Customizable, Self-Learnable Parameterized MT System: The Next Generation," (invited paper) Proceedings of the Machine Translation Summit VII '99, pp. 182-188, Natl. Univ. of Singapore, Singapore, Sept. 13-17, 1999.

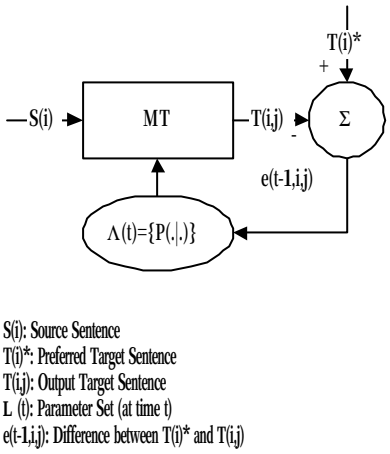

Figure 3: Including User Feedback in a Parameterized MT for User Adaptation

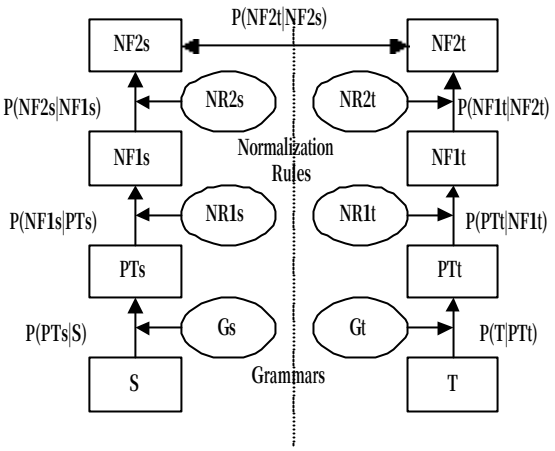

Figure 4: Two-way Training for Automatically Acquiring the Translation Knowledge

Figure 3 suggests that we can adjust the set of system parameters toward special user preference by feeding a set of source sentences S(*i*) to the parameterized MT system, and comparing it's output target sentence T(*i*, *j*) with the preferred sentences T(*i*)*. Should there be any discrepancy e(*t-1*,*i*,*j*) between the preferred target sentence and the output (resulted from using the current set of parameters), the parameters could be adjusted using some well-developed adaptive learning methods (e.g., [Amari 67]). To reduce the cost for constructing the system, the initial set of parameters could be trained with some unsupervised training methods, such as an EM algorithm [Dempster 77]. Therefore, the user preference can easily be satisfied by the proposed architecture.

5 Two-way Training for Knowledge Acquisition

Given the proposed architecture, what remains to be resolved is then a training method which can automatically learn those parameters from the corpus, and, at the same time, prevent the generated target sentences from being affected by the source language. For these purposes, a method that is different from conventional one-way training process, as mention previously, must be adopted. Such a training process can be easily implemented as shown in Figure 4. Briefly speaking, the two-way training process will prepare a bilingual corpus which had not been annotated with various normal forms (i.e., intermediate representations) of the source and target sentences. To reduce the training cost, the parameters will be obtained using an unsupervised method, like the EM algorithm (or Viterbi training [Rabiner 93]).

Figure 5 shows the idea on how to train the system parameters in such an unsupervised manner. For instance, given the English sentence "this is a
its counterpart in Chinese, we can parse the two sentences (with their respective analysis grammars) to acquire all possible candidate analyses, which consist of various normal forms (i.e., intermediate representations) of the source and target sentences. The training process then try to find the best match between the deepest structures (i.e., the NF2 structures in Figure 2) of the corresponding sentences based on a specified objective scoring functions. Because the parameters associated with the generation of the target sentences are acquired independent of the source language, and the intermediate representations for the preferred target sentences are within the grammar of the target language, it could be expected that the generated target sentences will be less affected by the source language using such a two-way training process.

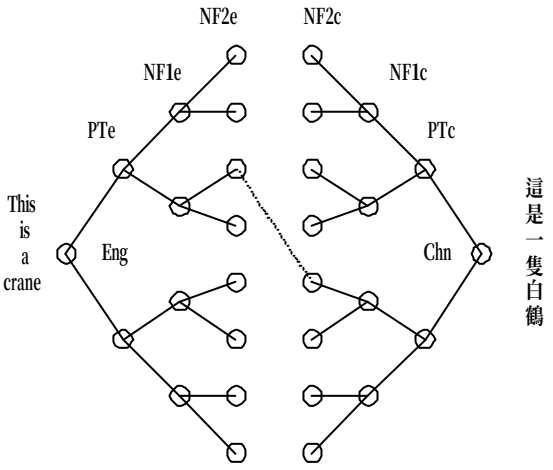

Figure 5: An Example for Learning Translation Knowledge with the Two-Way Training

The unsupervised learning actually goes through a sequence of re-estimation steps. Taking the Viterbi training as an example (which is designed to learn the

simplified case of hard-labeling), the system will randomly select (or guessed with other relevant knowledge) a matching between the deepest normal forms from each side initially. Afterwards, with such an initial guess, we can get a set of initial parameters by using the Maximum Likelihood Estimation method, and then use this initial set to further find a better mapping. This process repeats like a typical EM algorithm does, until the likelihood value converge to a local maximum. Once the best mapping is acquired, we can then estimate the parameters required for acquiring the best translations from the English sentences to the preferred Chinese sentences (or vice versa). Furthermore, in the translation process, the target sentence will be generated based on the deep structures of the target language (instead of a modified version of the deep structures of the source language). We therefore can expect a target sentence that is less affected by the styles specific to the source language.

Also, note that the source sentence and the most preferred target sentence are kept unchanged through the whole training process. In other words, we are proposing a two-end constrain optimization process, where the parameters are tuned toward producing the best target translation. Such a constrain-satisfaction scheme is more likely to lead us to a better local maximum in the parameter space, because each language will impose constrains on the possible structures of the other side, thus reducing the degree of freedom of the relevant parameters ([Dagon 91]). Using an instance in the English-Chinese bilingual corpus as an example, the English word "crane" have at least two senses: one is the "bird-sense" and the other is the "machinery-sense". If the unsupervised learning mechanism for semantic sense disambiguation is only applied to an English monolingual corpus, then it would be very difficult to know the human preference, and thus the parameters are likely to converge to a wrong local maximum point. On the other hand, if its Chinese translation is also given, as the Chinese term "Bai-Heh" for the "bird-sense" has no ambiguity in the Chinese part[3], the system then will know that the word "crane" in the English sentence, in this case, should have -sense" attached to it.

One obvious advantage of this two-way training mechanism is the customization capability. We can prepare a large balance corpus (for covering general patterns) and a set of domain-specific corpora (each for one specific domain). Each special-purpose MT system, for one specific domain, is obtained by associating it with a specific set of parameters, which are acquired from the mixture of the balance corpus and the domain-specific corpus, through the above two-way training mechanism. Thus the domain customization (and even the company customization) could be achieved in a very quick and cheap way. In this way, we can expect a new generation MT architecture, which is able to satisfy the requirements in putting MT systems into the translation market of the real world.

[3] Even it has ambiguities in Chinese language, the distribution in semantic classes of those ambiguities would be quite different from that for the "crane" in the English side. Therefore, it is easier for the unsupervised learning to find the most possible mapping. On the other hand, if the corresponding Chinese term has the same distribution in the semantic sense as its English counter part, then the advantage of using a bilingual corpus must be achieved by the context around this word. In fact, this is the issue related to what is "learnable" or "identifiable" [Duda 73].

6 Concluding Remarks

In this paper, we had proposed an architecture for the next generation MT, which is characterized by a probabilistic parameterized system, and thus, can be easily customized for different domains and users (which implies a larger market size). It is also featured by the capability to include the user feedback for tailoring the system parameters toward a particular user's preference.

Besides, an automatic training method, called two-way training, is also proposed in this paper to get a set of generation parameters that is independent of the source language. Because of the automatic training nature, the costs for acquiring the mass amount of translation knowledge could be significantly reduced, and the user preference could be well adapted automatically. The two-way training nature further ensures that we can expect high quality source-language-independent output, and get better user satisfaction. Such reduction in cost and increase in user satisfaction will be the key to successfully enlarge the market share of the MT services. We believed that such a new architecture and training method will play an important role in the next generation MT systems.

References

[Amari 67] Amari, S., (1967). "A theory of adaptive pattern classifiers", *IEEE Trans. on Electronic Computers*, vol. EC-16, no. 3, pp. 299-307, June 1967.

[Dagon 91] Ido Dagan, Alon Itai, and Ulrike Schwall, (1991). "Two Languages Are More Informative Than One", *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pp. 130-137, 18-21 June 1991, UC Berkeley, CA, USA.

[Dempster 77] Dempster, A. P., N. M. Laird and D. R. Rubin, (1977) "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society*, 39 (B), pp. 1-38, 1977.

[Duda 73] Duda, O.R. and P.E. Hart, (1973) *Pattern Classification and Scene Analysis*, John Wiley and Sons Inc., USA.

[Hutchins 86] Hutchins, W. J., (1986). "Machine Translation: Past, Present, Future," Published by Ellis Horwood Limited, Distributed by John Wiley&Sons, USA, 1986.

[Rabiner 93] Rabiner, L., B.-H. Juang, (1993). *Fundamentals of Speech Recognition*, Prentice Hall International, 1993.

[Su 92] Su, K.-Y., M.-W. Wu and J.-S. Chang, (1992). "A New Quantitative Quality Measure for Machine Translation Systems," *Proceedings of COLING-92*, vol. II, pp. 433--439, 14th Int. Conference on Computational Linguistics, Nantes, France, 1992.

[Su 95] Su, K.-Y., J.-S. Chang and Y. -L. Una Hsu, (1995). "A Corpus-Based Statistics-Oriented Two-Way Design for Parameterized MT Systems: Rationale, Architecture and Training Issues", *Proceedings of TMI-95*, vol 2, pp. 334-353, Centre for Computational Linguistics, Katholieke Universiteit Leuven, Leuven, Belgium, July 5-7, 1995.

[Su 96] Su, Keh-Yih, Tung-Hui Chiang, and Jing-Shin Chang, (1996). "An Overview of Corpus-Based Statistics-Oriented (CBSO) Techniques for Natural Language Processing," *International Journal of Computational Linguistics and Chinese Language Processing*, pp. 101-157, Academia Sinica, Taipei, ROC.