Combining translation models in statistical machine translation

Jesús Andrés-Ferrer
PRHLT Group
UPV
jandres@dsic.upv.es

Ismael García-Varea
PRHLT Group
UCML
ivarea@info-ab.uclm.es

Francisco Casacuberta
PRHLT Group
UPV
fcn@dsic.upv.es

Abstract

Originally, statistical machine translation was based on the use of the "noisy channel" approach. However, many of the current and successful statistical machine translation systems are based on the use of a direct translation model or even on the use of a log-linear combination of serveral direct and inverse translation models. An attempt to justify the use of these heuristic systems was proposed within the framework of maximum entropy.

We present a theoretical justification under the decision theory framework. This theoretical frame entails new methods for increasing the performance of the systems combining translation models. We propose new and more powerful translation rules that also fit within this theoretical framework. The most important theoretical properties developed in the paper are experimentally studied through a simple translation task.

1 Introduction

Machine Translation (MT) deals with the problem of automatically translating a sentence (**f**) from a source language[1] ($\mathbf{F}^*$) into a sentence (**e**) from a target language ($\mathbf{E}^*$). Obviously, these two languages are supposed to have a very complex set of rules involved in the translation process that cannot be properly enumerated into a computer system. According to this, many authors have embraced a statistical approach to the MT problem, where the only source of information is a parallel corpus of source-to-target translated sentences.

Brown et al. (1993) approached the problem of MT from a purely statistical point of view. In this approach, the MT problem is analysed as a classical pattern recognition problem using the well-known Bayes' classification rule (Duda et al., 2000). Therefore, statistical machine translation (SMT) is a classification task where the set of classes is the set of all sentences of the target language ($\mathbf{E}^*$), i.e. every target string ($\mathbf{e} \in \mathbf{E}^*$) is regarded as a possible translation for the source language string (**f**). The goal of the translation process in statistical machine translation can be formulated as follows: a source language string **f** is to be translated into a target language string **e**[2]. Then the system searches the target string ($\hat{\mathbf{e}}$) with maximum a-posteriori probability $p(\mathbf{e}|\mathbf{f})$:

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}\in\mathbf{E}^*}\{p(\mathbf{e}|\mathbf{f})\} \qquad (1)$$

where $p(\mathbf{e}|\mathbf{f})$ can be approached through a *direct* statistical translation model. Eq. (1) has proved to be the optimal

[1] $\mathbf{F}^*$ is the set of all possible strings with a finite length on the lexicon **F**.

[2] We will refer to $p(\mathbf{e}|\mathbf{f})$ as a direct statistical translation model and to $p(\mathbf{f}|\mathbf{e})$ as an inverse statistical translation model.

decision/classification rule under some assumptions and is called the optimal Bayes' classification rule (obviously assumes that the actual probability distribution $p(\mathbf{e}|\mathbf{f})$ is known). Applying the Bayes' theorem to Eq. (1), the following rule is obtained:

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}\in\mathbf{E}^*}\{p(\mathbf{e}) \cdot p(\mathbf{f}|\mathbf{e})\} \tag{2}$$

Eq. (2) implies that the system has to search the target string ($\hat{\mathbf{e}}$) that maximises the product of both, the target language model $p(\mathbf{e})$ and the inverse string translation model $p(\mathbf{f}|\mathbf{e})$. Thus, the Bayes' classification rule provides the *inverse translation rule* (ITR), which is also called "the fundamental equation of SMT". Again, this rule is optimal if the actual models are known. Nevertheless, using this rule implies, in practice, changing the distribution probabilities as well as the models through which the probabilities are approached. This is exactly the advantage of this approach, as it allows the modelling of the direct translation probability ($p(\mathbf{e}|\mathbf{f})$) with two models: an inverse translation model that approximates $p(\mathbf{f}|\mathbf{e})$; and a language model that approximates $p(\mathbf{e})$.

This approach has a strong practical drawback: the search problem[3]. This search is known to be an NP-hard problem (Knight, 1999; Udupa and Maji, 2006). However, several search algorithms have been proposed in the literature to solve this ill-posed problem efficiently (Brown and others, 1990; Wang and Waibel, 1997; Yaser and others, 1999; Germann and others, 2001; Jelinek, 1969; García-Varea and Casacuberta, 2001; Tillmann and Ney, 2003).

In order to alleviate this drawback, many of the current SMT systems (Och et al., 1999; Och and Ney, 2004; Koehn et al., 2003; Zens et al., 2002) have proposed the use of the *direct translation rule* (DTR):

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}\in\mathbf{E}^*}\{p(\mathbf{e}) \cdot p(\mathbf{e}|\mathbf{f})\} \tag{3}$$

which can be seen as an heuristic version of the ITR (Eq. (2)), where $p(\mathbf{f}|\mathbf{e})$ is substituted by $p(\mathbf{e}|\mathbf{f})$. This rule allows an easier search algorithm for some of the translation models.

Although the DTR has been widely used, its statistical theoretical foundation has not been clear for long time, as it seemed to be against the Bayes' classification rule if an *asymmetric model*[4] is used for modelling the translation probability. Other authors (Andrés-Ferrer et al., 2007) have provided an explanation of its use within decision theory. In this work, we expand that theory to other translation models and other loss functions, providing a general framework to combine translation systems.

Some of the current SMT systems (Och and Ney, 2004; Marino et al., 2006) use a log-linear combination of statistical models to approximate the direct translation distribution:

$$p(\mathbf{e}|\mathbf{f}) \approx \frac{\exp\left[\sum_{m=1}^{M}\lambda_m h_m(\mathbf{f},\mathbf{e})\right]}{\sum_{\mathbf{e}'}\exp\left[\sum_{m=1}^{M}\lambda_m h_m(\mathbf{f},\mathbf{e}')\right]} \tag{4}$$

where h_m is a logarithmic statistical model that approximates a probability distribution (i.e. translation or language probabilities).

The paper is organised as follows: section 2 summarises the Bayes' decision theory. Section 3 tackles SMT under the decision theory framework. Finally, section 4 demonstrates in practice the theoretical ideas explained in the paper. Conclusions are condensed in the section 5.

2 Bayes Decision Theory

A classification problem such as the SMT problem can be seen as an instance of a Decision Problem (DP). From this point of view, a classification problem is composed of three different items:

1. A set of *Objects* ($\mathcal{X}$) the system might observe and has to classify (i.e., translate).

2. A set of classes ($\Omega = \{\omega_1, \ldots, \omega_C\}$) in which the system has to classify each observed object $\mathbf{x} \in \mathcal{X}$.

[3] The method for solving the maximisation (or the search) of the optimal $\hat{\mathbf{e}}$ in the set $\mathbf{E}^*$, i.e. $\arg\max_{\mathbf{e}\in\mathbf{E}^*}$

[4] Given two sentences $\mathbf{e}$ and $\mathbf{f}$ from the target and source language: a *symmetric* model assigns the same probability to $p(\mathbf{e}|\mathbf{f})$ and to $p(\mathbf{f}|\mathbf{e})$; and an *asymmetric* model does not.

3. A *Loss function* ($\mathrm{l}(\omega_k|\mathbf{x},\omega_j)$). This function evaluates the loss of classifying an observed object $\mathbf{x}$ in a class, $\omega_k \in \Omega$, knowing that the *optimal class* for the object $\mathbf{x}$ is $\omega_j \in \Omega$.

Therefore, when an object $x \in \mathcal{X}$ is observed in a classification system, the system chooses the "correct" class from all possible classes (Ω). The term "correct" is used in the sense of the action that minimises the loss in which the system could incur if it makes an error, according to the loss function. For reasons of simplicity, the 0-1 *loss function* is usually assumed, i.e.:

$$\mathrm{l}(\omega_k|\mathbf{x},\omega_j) = \begin{cases} 0 & \omega_k = \omega_j \\ 1 & \text{otherwise} \end{cases} \tag{5}$$

This loss function does not penalise the correct class, nevertheless it does not distinguish between the importance of classifying an object in a specific wrong class or in another wrong class. Therefore, the penalty of classifying the object $\mathbf{x}$ in the class ω_i or ω_j is the same. This is only sensible in some small and simple cases. For example, if the set of classes is large, or even infinite (but still enumerable), then it is not very appropiate to penalise all wrong classes the same. Note that in this case it is impossible to define a uniform distribution over the classes. This implies that there are classes that have a very small probability, and then it does not make sense to define a uniform loss function for those classes. Instead, it is better to penalise the zones where the probability is high.

In order to build a classification system the *classification function* must be defined, say $\mathrm{c}: \mathcal{X} \to \Omega$. The class provided by the classification function may not be the correct class. Thereby, the classification function yields an error or risk, the so-called *Global Risk*,

$$R(\mathrm{c}) = \mathrm{E}_{\mathbf{x}}[R(\mathrm{c}(\mathbf{x})|\mathbf{x})] = \int_{\mathcal{X}} R(\mathrm{c}(\mathbf{x})|\mathbf{x})\,\mathrm{p}(\mathbf{x})d\mathbf{x} \tag{6}$$

where $R(\omega_k|\mathbf{x})$ (with $\omega_k = \mathrm{c}(\mathbf{x})$) is the *Conditional Risk given* $\mathbf{x}$, i.e. the expected loss of classifying in the class determined by the decision function. This *Conditional Risk* is expressed as follows:

$$R(\omega_k|\mathbf{x}) = \sum_{\omega_j \in \Omega} \mathrm{l}(\omega_k|\mathbf{x},\omega_j)\,\mathrm{p}(\omega_j|\mathbf{x}) \tag{7}$$

The well-known *Bayes' classification rule* is the rule that minimises the Global Risk. Moreover, as minimising the Conditional Risk for each object ($\mathbf{x}$) is a sufficient condition to minimise the Global Risk, without loss of generality we can say that the optimal *Bayes classification rule* is the rule that minimises the Conditional Risk, i.e.:

$$\hat{\mathrm{c}}(\mathbf{x}) = \arg\min_{\omega \in \Omega} R(\omega|\mathbf{x}) \tag{8}$$

Loss functions that are more appropriate than the 0-1 can be designed. If we only assume that the loss of correctly classifying an object is 0, then a very general loss function is obtained:

$$\mathrm{l}(\omega_k|\mathbf{x},\omega_j) = \begin{cases} 0 & \omega_k = \omega_j \\ \epsilon(\mathbf{x},\omega_k,\omega_j) & \text{otherwise} \end{cases} \tag{9}$$

In the case of Eq.(9), the optimal Bayes' classifier is given by:

$$\hat{\mathrm{c}}(\mathbf{x}) = \arg\min_{\omega_k \in \Omega} \sum_{\omega_j \neq \omega_k} \epsilon(\mathbf{x},\omega_k,\omega_j)\,\mathrm{p}(\omega_j|\mathbf{x}) \tag{10}$$

Note that in order to perform the search for the optimal class $\hat{\mathrm{c}}(\mathbf{x})$ it is necessary to find the class ω_k, for which the sum over all the remaining classes ω_j is mimimun. This requires a computation time[5]of $O(|\Omega|^2)$. This cost can be prohibitive in some problems. For instance, in machine translation, the set of classes is exponential with the length of the sentence. In this case, having to compute the sum for each class is a practical problem that can ruin the advantages obtained by using a more appropriate loss function.

In this sense, there is a particular set of loss functions of the form of Eq. (9), that preserves the simplicity of the optimal classification rule for the 0-1 loss function. If ω_k is the class proposed by the system and ω_j is the correct class

[5] Note that we are assuming that the cost of evaluating $\epsilon(\mathbf{x},\omega_k,\omega_j)$ and $\mathrm{p}(\omega_j|\mathbf{x})$ is costant in time

that the system should choose (ω_k is expected to be equal to ω_j) the following loss function $\mathrm{l}(\omega_k|\mathbf{x}, \omega_j)$ preserves this simplicity:

$$\mathrm{l}(\omega_k|\mathbf{x}, \omega_j) = \begin{cases} 0 & \omega_k = \omega_j \\ \epsilon(\mathbf{x}, \omega_j) & \text{otherwise} \end{cases} \tag{11}$$

where $\epsilon(\cdot)$ is a function depending on the object ($\mathbf{x}$) and the correct class (ω_j) but not depending on the wrong class proposed by the system (ω_k). This function must verify that $\sum_{\omega_j \in \Omega} \mathrm{p}(\omega_j|\mathbf{x})\, \epsilon(\mathbf{x}, \omega_j) < \infty$; and it evaluates the loss function when the system fails.

In such cases, it can be easily proved that the Conditional Expected Risk is:

$$R(\omega_k|\mathbf{x}) = \mathrm{S}(\mathbf{x}) - \mathrm{p}(\omega_k|\mathbf{x})\, \epsilon(\mathbf{x}, \omega_k) \tag{12}$$

where $\mathrm{S}(\mathbf{x}) = \sum_{\omega_j \in \Omega} \mathrm{p}(\omega_j|\mathbf{x})\, \epsilon(\mathbf{x}, \omega_j)$ and $\mathrm{S}(\mathbf{x}) < \infty$, i.e. the weighted sum over all possible classes converges to a finite number which only depends on $\mathbf{x}$. Therefore, $\epsilon(\cdot)$ is restricted to functions that hold the previous finiteness property.

As a result, the classification rule is very similar to the optimal Bayes' classification rule for the 0-1 loss function and simplifies to the following equation (Andrés-Ferrer et al., 2007):

$$\hat{\mathrm{c}}(\mathbf{x}) = \arg\max_{\omega \in \Omega} \{\mathrm{p}(\omega|\mathbf{x})\, \epsilon(\mathbf{x}, \omega)\} \tag{13}$$

It is worth noting that the computational time[6] needed to sovle the search of the optimal class in Eq. (13). is $O(|\Omega|)$.

In conclusion, for each loss function there exists a different optimal Bayes' classification rule, specifically using a loss function like the one in Eq. (11) yields one of the simplest optimal classification rules, Eq. (13).

3 Statistical Machine Translation

SMT is a specific instance of a classification problem where the set of possible classes is the set of all the possible sentences that might be written in a target language, i.e. $\Omega = \mathbf{E}^*$. Likewise, the objects to be classified[7]are sentences of a source language, i.e. $\mathbf{f} \in \mathbf{F}^*$.

In a SMT system, the Bayes' classification rule is Eq. (2). As stated above, this classification rule can be obtained by using the 0-1 loss function:

$$\hat{\mathbf{e}} = \hat{\mathrm{c}}(\mathbf{f}) = \arg\max_{\omega_k \in \Omega} \{\mathrm{p}(\omega_k|\mathbf{f})\} \tag{14}$$

where $\omega_k = \mathbf{e}_k$. This loss function is not particularly appropriate when the number of classes is huge as occurs in SMT problems. Specifically, if the correct translation for the source sentence $\mathbf{f}$ is $\mathbf{e}_j$, and the hypothesis of the translation system is $\mathbf{e}_k$; using the 0-1 loss function (Eq. (5)) has the consequence of penalising the system in the same way, independently of which translation ($\mathbf{e}_k$) the system proposes and which is the correct translation ($\mathbf{e}_j$) for the source sentence ($\mathbf{f}$).

3.1 Quadratic loss functions

Equation (9) produces search algorithms which have a quadratic cost depending on the size of the set of classes. As stated above, machine translation can be understood as a classification problem with a huge set of classes. Hence, these loss functions yield difficult search algorithms. There are some works that already have explored this kind of loss functions (Ueffing and Ney, 2004; R. Schlüter and Ney, 2005).

The more appealing application of this loss functions is the use of a metric loss function (R. Schlüter and Ney, 2005). For instance, in machine translation one widespread metric is the WER (see Section 4 for a definition), since the loss function in Equation (9) depends on both, the proposed translation and the reference translation, the WER can be used as loss function (Ueffing and Ney, 2004). Nevertheless, due to the high complexity, the use of these quadratic and interesting loss functions, is only feasible in constrained situations like n-best lists (Kumar and Byrne, 2004).

[6] Note that we are assuming that the cost of evaluating $\epsilon(\mathbf{x}, \omega_j)$ and $\mathrm{p}(\omega_j|\mathbf{x})$ is costant in time

[7] In this context to classify an object $\mathbf{f}$ in the class ω_k is a way of expressing that $\mathbf{e}_k$ is the translation of $\mathbf{f}$.

Another interesting loss function would be the one obtained by introducing a kernel as the loss function in Equation (9):

$$\mathrm{l}(\mathbf{e}_k|\mathbf{f},\mathbf{e}_j) = \begin{cases} 0 & \mathbf{e}_k = \mathbf{e}_j \\ \mathcal{K}_n(\mathbf{e}_k,\mathbf{e}_j) & \text{otherwise} \end{cases} \tag{15}$$

with

$$\mathcal{K}_n(\mathbf{e}_k,\mathbf{e}_j) = \sum_{\mathbf{u}\in E^n} |\mathbf{e}_j|_{\mathbf{u}} |\mathbf{e}_k|_{\mathbf{u}} \tag{16}$$

where $|\mathbf{e}|_{\mathbf{u}}$ stands for the number of occurrences of the sequence of n words $\mathbf{u}$ inside the sentence $\mathbf{e}$ (Cortes et al., 2005).

3.2 Linear loss function

Equation (11) produces search algorithms which have a linear cost depending on the size of the set of classes. For instance, a more suitable loss function than the 0–1 loss, can be obtained using Eq. (11) with $\epsilon(\mathbf{f},\mathbf{e}_j) = p(\mathbf{e}_j)$:

$$\mathrm{l}(\mathbf{e}_k|\mathbf{f},\mathbf{e}_j) = \begin{cases} 0 & \mathbf{e}_k = \mathbf{e}_j \\ p(\mathbf{e}_j) & \text{otherwise} \end{cases} \tag{17}$$

This loss function seems to be more appropriate than the 0-1. This is due to the fact that if the system makes an error translating a set of source sentences, this loss function tries to force the system to fail in the source sentence ($\mathbf{f}$) whose correct translation[8] ($\mathbf{e}_j$) is one of the least probable in the target language. Thus, the system will fail in the least probable translations, whenever it gets confused; and therefore, the *Global Risk* will be reduced.

In addition, it is easy to prove (using Eq. (13)) that this loss function leads to the Direct Translation Rule in Eq. (3). Then, the DTR should work better than the ITR, from a theoretical point of view.

Nevertheless, the statistical approximations employed for modelling translation probabilities might not be symmetric, as is the case with IBM Models (Brown and other, 1993). Thus, the model error, could be more important than the advantage obtained from the use of a more appropriate loss function. Therefore, it seems a good idea to use the direct rule in the equivalent inverse manner so that the translation system will be the same and then these asymmetries will be reduced. By simply applying the Bayes' theorem to Eq. (3), we obtain the equivalent rule:

$$\hat{\mathbf{e}} = \operatorname*{arg\,max}_{\mathbf{e}\in\mathbf{E}^*} \left\{ p(\mathbf{e})^2 p(\mathbf{f}|\mathbf{e}) \right\} \tag{18}$$

The difference between the Eq (3) and Eq (18) can be used to measure the asymmetries of the translation models.

An alternative function to the proposed in Eq (17) is the loss function in Eq. (11) with $\epsilon(\mathbf{f},\mathbf{e}_j) = p(\mathbf{f},\mathbf{e}_j)$:

$$\mathrm{l}(\mathbf{e}_k|\mathbf{f},\mathbf{e}_j) = \begin{cases} 0 & \mathbf{e}_k = \mathbf{e}_j \\ p(\mathbf{f},\mathbf{e}_j) & \text{otherwise} \end{cases} \tag{19}$$

which leads to:

$$\hat{\mathbf{e}} = \operatorname*{arg\,max}_{\mathbf{e}\in\mathbf{E}^*} \left\{ p(\mathbf{f},\mathbf{e}) p(\mathbf{e}|\mathbf{f}) \right\} \tag{20}$$

Equation (20) is able to provide several optimal classification rules depending on which approximation is used to model the joint probaility ($p(\mathbf{f},\mathbf{e})$). The most important rule produced by this function is the *Inverse and Direct translation rule (I&DTR)*, which is expressed by the following equation:

$$\hat{\mathbf{e}} = \operatorname*{arg\,max}_{\mathbf{e}\in\mathbf{E}^*} \left\{ p(\mathbf{e}) p(\mathbf{f}\,|\,\mathbf{e}) p(\mathbf{e}\,|\,\mathbf{f}) \right\} \tag{21}$$

The interpretation of this rule is a refinement of the direct translation rule. In this case, if the system makes a mistake it is done in the least probable pairs $(\mathbf{f},\mathbf{e})$ in terms of $p(\mathbf{e},\mathbf{f})$.

More interesting loss functions can be obtained using information theory. For instance, we can penalise the system by the *remaining information*. That is, if we knew $p(\mathbf{e})$, then the information associated with a target sentence $\mathbf{e}_j$ would be $-\log(p(\mathbf{e}_j))$. The remaining information, or the information that the system has learnt when it fails is given by $-\log(1-p(\mathbf{e}_j))$. Hence, the system can be penalised with this score:

$$\mathrm{l}(\mathbf{e}_k|\mathbf{f},\mathbf{e}_j) = \begin{cases} 0 & \mathbf{e}_k = \mathbf{e}_j \\ -\log(1-p(\mathbf{f},\mathbf{e}_j)) & \text{otherwise} \end{cases} \tag{22}$$

[8] Here lies the importance of distinguishing between the translation proposed by the system ($\mathbf{e}_k$) and the correct translation ($\mathbf{e}_j$) of the source sentence($\mathbf{f}$).

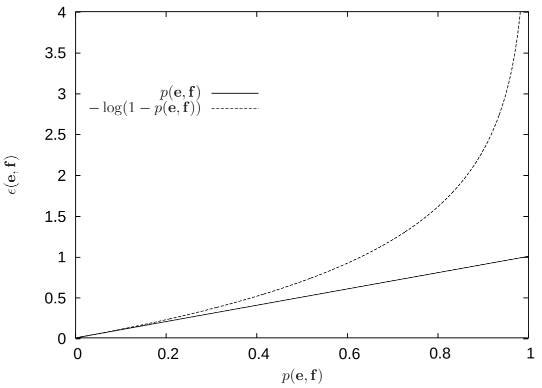

Figure 1: The information of the contrary event, or the remaining information.

Figure 1, shows the remaining information of a probability function. Note that the remaining information has a singularity at 1, i.e. if the system has not been able to learn a sure event, which has probability of 1, then the loss is infinity. Note that this loss can be defined for any probability such as p($\mathbf{e}$) or p($\mathbf{x}, \mathbf{e}$).

Some works (Och and Ney, 2004; Marino et al., 2006), explore the idea of using maximum entropy models to design a translation system, obtaining in this way a translation rule of the form of:

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e} \in \mathbf{E}^*} \sum_{m=1}^{M} \lambda_m h_m(\mathbf{f}, \mathbf{e}) \qquad (23)$$

where h_m is a logarithmic statistical model that approximates a probability distribution (i.e. translation or language probabilities).

The Eq (23) can be analysed from a Bayes' decision theory frame. Into this scope, what the log-linear systems are doing is to use the loss function in Eq (11) with:

$$\epsilon(\mathbf{f}, \mathbf{e}) = p(\mathbf{e} \,|\, \mathbf{f})^{-1} \prod_{m=1}^{M} f_m(\mathbf{f}, \mathbf{e})^{\lambda_i} \qquad (24)$$

where $f_m(\mathbf{f}, \mathbf{e}) = \exp[h_m(\mathbf{f}, \mathbf{e})]$.

From the decision theory, the log-linear models learn the best loss function among a family of loss functions. This family is defined by a vector of hyperparameters ($\boldsymbol{\lambda}_{\mathbf{1}}^{\boldsymbol{M}}$):

$$\left\{ p(\mathbf{e} \,|\, \mathbf{f})^{-1} \prod_{m=1}^{M} f_m(\mathbf{f}, \mathbf{e})^{\lambda_i} \;\middle|\; \forall \lambda_i \right\} \qquad (25)$$

In order to perform the optimisation, firstly the f_m functions (usually an exponential functions of probability distributions) are estimated using maximum likelihood (or some other estimation technique). Secondly, the ME algorithm (Berger et al., 1996) is used to find the optimal weights or hyperparameters λ_i, i.e., the ME algorithm is used to find the optimal loss function among all the possible functions in the family.

Some works explore the idea of using these hyperparameters to reduce the evaluation error metric, such as the Bleu (Papineni et al., 2001). For instance, in Och (2003), some improvements were reported when estimating the hyperparameters $\boldsymbol{\lambda}$ in accordance with the evaluation metric.

4 Experimental Results

The aim of this section is to demonstrate with practical results, how to use the theory stated in the work to improve the performance of a translation system. Obtaining a state-of-art system is out of scope of this paper. In this way, the previously stated properties will be analysed in practice with a simple translation model. In other works, some of the loss functions presented here has been analysed using state-of-art models, phrase-based models, (Andrés-Ferrer et al., 2007)

Before starting the section we need to define two new concepts (Germann and others, 2001). When a SMT system proposes a wrong translation, this is due to two reasons: the suboptimal search algorithm which has not been able to compose a good translation; or the model which is not able to make up a good translation (and so is unable to find it). Then we will say that a translation error is a *search error (SE)* if the probability of the proposed translations is less than the reference translation; otherwise we will say that it is a *model error*, i.e. if the probability of the proposed translations is greater than the reference translation.

We use the IBM Model 2 (Brown and other, 1993) and the corresponding search algorithms to design the experiments of this work. That choice was motivated by several reason. Firstly, the simplicity of the translation model allows to obtain a good estimation of the model parameters. Secondly, there are several models that are initialised using the alignments and dictionaries of the IBM model 2. Finally, the search problem can be solved exactly using dynamic programming for the DTR.

In order to train the IBM Model 2 we used the standard tool *GIZA++* (Och, 2000). We re-implemented the algorithm presented in (García-Varea and Casacuberta, 2001) to perform the search process in translation for the ITR. Even though this search algorithm is not optimal, we set the parameters to minimise the search errors, so that all the errors should be model errors. In addition we implemented the corresponding version of this algorithm for the DTR and for the I&DTR. All these algorithms were developed by dynamic programming. For the I&DTR, we implemented two versions of the search: one guided by the direct model (a non-optimal search algorithm, namely I&DTR-D) and the other guided by the inverse translation model (which is also non-optimal but more accurate, namely I&DTR-I). Due to the length constraint of the article, the details of the algorithms are omitted.

We selected the Spanish-English TOURIST task (Amengual et al., 1996) to carry out the experiments reported here. The Spanish-English sentence pairs correspond to human-to-human communication situations at the front-desk of a hotel which were semi-automatically produced. The parallel corpus consisted of 171,352 different sentence pairs, where 1K sentences were randomly selected from testing, and the rest (in sets of exponentially increasing sizes: 1K, 2K, 4K, 8K, 16K, 32K, 64K, 128K and 170K sentences pairs) for training. The basic statistics of this corpus are shown in Table 1. All the figures show the confidence interval at 95%.

In order to evaluate the translation quality, we used the following well-known automatically computable measures:

1. *Word Error Rate* (WER):Word Error Rate is the minimum number (in %) of deletions, insertions, and substitutions that are necessary to transform the translation proposed by the system into the reference translation.

2. *Sentence Error Rate* (SER): Sentence Error Rate is the number (in %) of sentences that differs from the reference translations.

3. *BiLingual Evaluation Understudy* (BLEU): it is based on the n-grams of the hypothesized translation that occur in the reference translations. In this work, only one reference translation per sentence was used. The BLEU metric ranges from 0.0 (worst score) to 1.0 (best score) (Papineni et al., 2001):

	Test Set		Train Set	
	Spa	Eng	Spa	Eng
sentences	1K		170K	
avg. length	12.7	12.6	12.9	13.0
vocabulary	518	393	688	514
singletons	107	90	12	7
perplexity	3.62	2.95	3.50	2.89

Table 1: Basic statistics of the Spanish-English TOURIST task.

Figure 2 shows the differences in terms of the WER among all the mentioned forms of the DTR: “IFDTR” (Eq. 18), “DTR” (Eq. 3), and “DTR-N” (Normalised Length version of DTR). Note the importance of the model asymmetry in the obtained results. The best results were the ones obtained using the inverse form of the DTR. The normalised version was developed due to the fact that the IBM Model 2 (in its direct version) tries to provide very short translations. This behaviour is not surprising, since the only mechanism that the IBM Model 2 has to ensure that all sources words are translated is the length distribution. The length distribution usually allows the model to ommit the translation of a few words. Nevertheless, the “DTR” and “DTR-N” performed worse than the ITR (Table 2).

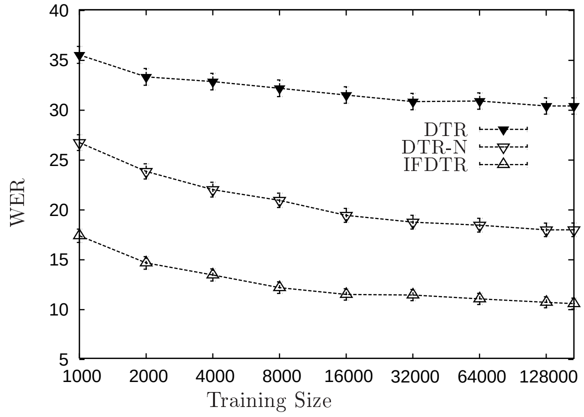

Figure 2: Asymmetry of the IBM Model 2 measured with the respect to the WER for the TOURIST test set for different training sizes.

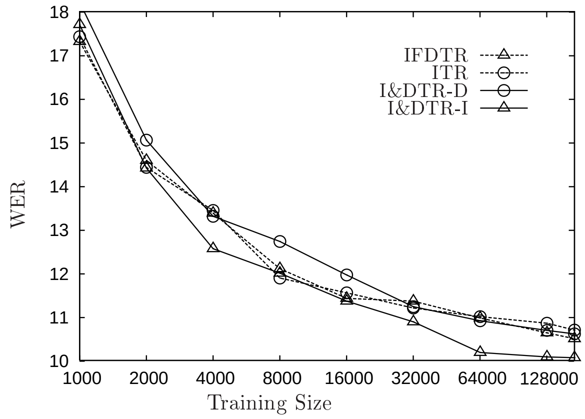

Figure 3: WER results for the TOURIST test set for different training sizes and different classification rules.

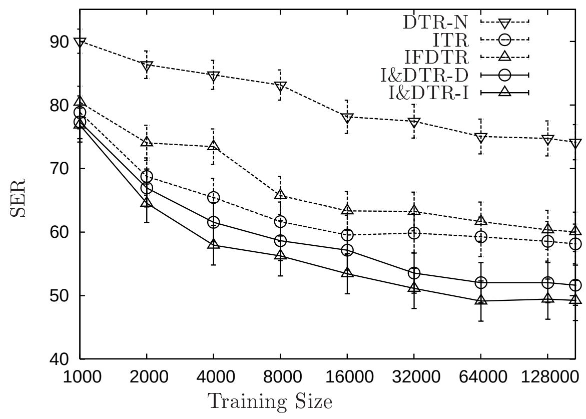

Figure 4: SER results for the TOURIST test set for different training sizes and different classification rules.

Model	WER	SER	BLEU	SE	T
I&DTR I	**10.0**	**49.2**	**0.847**	1.3	34
I&DTR D	10.6	51.6	0.844	9.7	2
IFDTR	10.5	60.0	0.837	2.7	35
ITR	10.7	58.1	0.843	1.9	43
DTR N	17.9	74.1	0.750	0.0	2
DTR	30.3	92.4	0.535	0.0	2

Table 2: Translation quality results with different translation rules for TOURIST test set for a training set of 170K sentences. Where T is the time expressed in seconds.

Figure 3 shows the results achieved with the most important rules. All the I&DTR obtain similar results to the ITR. Nevertheless, the non-optimal search algorithm guided by the direct model ("I&DTR-D") was an order of magnitude faster than the more accurate one ("I&DTR-I") and the ITR. The inverse form of the DTR ("IFDTR") behaved similarly to these, however improve the results reported by DTR. Therefore, there are no significant differences between the rules analysed in terms of WER. However, the execution times were significantly reduced by the direct guided search in comparison with the other searches. Table 2 shows these execution times and the figures with the maximum training size. Although the different search algorithms (based on loss functions) do not convey a significant improvement in WER. Note that the loss function only evaluates the SER, i.e. the loss function minimises the SER, and does not try to minimise the WER. Thus, changing the loss function, does not necessarily decrease the WER.

In order to support this idea, Figure 4 shows the analogous version of Figure 3 but with SER instead of WER. It should be noted that as the training size increases, there is a difference in the behaviour between the ITR and both I&DTR. Consequently, the use of these rules provides better SER, and this difference becomes statistically significant as the estimation of the parameters becomes better. In the case of the inverse form of the DTR ("IFDTR"), as the training size increases, the error tends to decrease and approximate the ITR error. However, the differences are not statistically significant and both methods are equivalent from this point of view.

In conclusion, there are two sets of rules: the first set is made up of IFDTR and ITR, and the second is composed by the two versions of the I&DTR. The first set reports worse SER than the the second set. However, the I&DTR guided with the direct model ("I&DTR-D") has many good properties in practice.

5 Conclusions

The analysis of the loss function is an appealing issue. The results of analysing different loss functions range from allowing to use metric loss functions such as BLEU, or WER; to proving the properties of some outstanding classification rules such as the direct translation rule, the inverse translation rule or even the maximumn entropy rule. For each different function $\epsilon(\mathbf{f}, \mathbf{e}_j, \mathbf{e}_k)$ in the general loss function of Eq. (9), there is a different optimal Bayes' rule. The point of using one specific rule is an heuristic and practical issue.

An interesting focus of study is the use of metrics such as BLEU, or WER; as the loss function. Nevertheless due to the high complexity, it is only feasible on constrained situations like n-best lists.

This work focuses on the study of loss functions that have a linear complexity and that are outstanding due to historical or practical reasons. In this sense, we have provided a theoretical approach based on decision theory which explains the differences and resemblances between the Direct and the Inverse Translation rules. This theoretical frame predicts an improvement (in terms of SER), an improvement that has been confirmed in practice.

In order to increase performance, we should find the best loss function with the form in Eq (9) or with the form in Eq (11). As future work, we will develop this idea into detail under the scope of functional optimisation. We also intend to analyse the practical behaviour of other loss functions such as the loss functions in Eq.(15) or the *remaining information* loss function.

Acknowledgements

This work has been supported by the EC (FEDER), the Spanish MEC under grant TIN2006-15694-CO2-01 and the Valencian "Conselleria d'Empresa, Universitat i Ciència" under grant CTBPRA/2005/004.

References

J.C. Amengual, J.M. Benedí, M.A. Castaño, A. Marzal, F. Prat, E. Vidal, J.M. Vilar, C. Delogu, A. di Carlo, H. Ney, and S. Vogel. 1996. Definition of a machine translation task and generation of corpora. Technical report d4, Instituto Tecnológico de Informática, September. ESPRIT, EuTrans IT-LTR-OS-20268.

J. Andrés-Ferrer, D. Ortiz-Martínez, I. García-Varea, and F. Casacuberta. 2007. On the use of different loss functions in statistical pattern recognition applied to machine translation. To appear in Pattern Recognition Letters.

A. L. Berger, Stephen A. Della Pietra, and V. J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–72, March.

P. F. Brown and other. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

P. F. Brown et al. 1990. A Statistical Approach to Machine Translation. *Computational Linguistics*, 16(2):79–85.

Corinna Cortes, Mehryar Mohri, and Jason Weston. 2005. A general regression technique for learning transductions. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 153–160, New York, NY, USA. ACM Press.

Richard O. Duda, Peter E. Hart, and David G. Stork. 2000. *Pattern Classification*. John Wiley and Sons, New York, NY, 2nd edition.

I. García-Varea and F. Casacuberta. 2001. Search algorithms for statistical machine translation based on dynamic programming and pruning techniques. In *Proc. of MT Summit VIII*, pages 115–120, Santiago de Compostela, Spain.

U. Germann et al. 2001. Fast decoding and optimal decoding for machine translation. In *Proc. of ACL01*, pages 228–235.

F. Jelinek. 1969. A fast sequential decoding algorithm using a stack. *IBM Journal of Research and Development*, 13:675–685.

Kevin Knight. 1999. Decoding complexity in word-replacement translation models. *Computational Linguistics*, 25(4):607–615.

P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL)*, Edmonton, Canada, May.

S. Kumar and W. Byrne. 2004. Minimum bayes-risk decoding for statistical machine translation.

J.B. Marino, R. E. Banchs, J.M. Crego, A. de Gispert, P. Lambert, J. A. R. Fonollosa, and M. R. Costa-jussà. 2006. N-gram-based machine translation. In *Computational Linguistics*, pages 527–549.

F.J. Och and H. Ney. 2004. The Alignment Template Approach to Statistical Machine Translation . *Computational Linguistics*, 30(4):417–449, December.

F. J. Och, Christoph Tillmann, and Hermann Ney. 1999. Improved alignment models for statistical machine translation. In *Proc. of the Joint SIGDAT Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 20–28, University of Maryland, College Park, MD, June.

F. J. Och. 2000. GIZA++: Training of statistical translation models. `http://www-i6.informatik.rwth-aachen.de/\~och/software/GIZA++.html`.

F. J. Och. 2003. Minimum error rate training in statistical machine translation. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 160–167, Morristown, NJ, USA. Association for Computational Linguistics.

Kishore A. Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. Bleu: a method for automatic evaluation of machine translation. Technical Report RC22176 (W0109-022), IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY, September.

V. Steinbiss R. Schlüter, T. Scharrenbach and H. Ney. 2005. Bayes risk minimization using metric loss functions. In *Proceedings of the European Conference on Speech Communication and Technology, Interspeech*, pages 1449–1452, Lisbon, Portugal, September.

Christoph Tillmann and Hermann Ney. 2003. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Computational Linguistics*, 29(1):97–133, March.

Raghavendra Udupa and Hemanta K. Maji. 2006. Computational complexity of statistical machine translation. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 25–32. Trento, Italy.

N. Ueffing and H. Ney. 2004. Bayes decision rules and confidence measures for statistical machine translation. In *EsTAL - Espa for Natural Language Processing*, pages 70–81, Alicante, Spain, October. Springer Verlag, LNCS.

Ye-Yi Wang and Alex Waibel. 1997. Decoding algorithm in statistical translation. In *Proc. of ACL'97*, pages 366–372, Madrid, Spain.

A. Yaser et al. 1999. Statistical Machine Translation: Final Report. Technical report, Johns Hopkins University 1999 Summer Workshop on Language Engineering, Center for Language and Speech Processing, Baltimore, MD, USA.

R. Zens, F.J. Och, and H. Ney. 2002. Phrase-based statistical machine translation. In *Advances in artificial intelligence. 25. Annual German Conference on AI*, volume 2479 of *Lecture Notes in Computer Science*, pages 18–32. Springer Verlag, September.