Estimation of Confidence Measures for Machine Translation

Alberto Sanchis, Alfons Juan, Enrique Vidal

Institut Tecnològic d'Informàtica
Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València
46071 València, SPAIN
{josanna,ajuan,evidal}@dsic.upv.es

Abstract

Confidence Estimation has been extensively used in Speech Recognition and now it is also being applied in Statistical Machine Translation. Its basic goal is to estimate a confidence measure for each word in a given hypothesis, in order to locate those words, if any, that are likely to be incorrectly recognised or translated. It can be seen as a two-class pattern recognition problem in which each hypothesized word is transformed into a feature vector and then classified as either correct or incorrect. This view provides a solid, well-known framework, within which accurate dichotomizers (two-class classifiers) can be derived. In this paper, we study the performance of certain pattern features along with a smoothed Naive Bayes dichotomizer. Good empirical results are reported on a translation task of technical manuals.

1 Introduction

Confidence Estimation (CE) has been extensively used in Speech Recognition (Wessel et al. 2001; Sanchis et al. 2004), and now it is also being applied in Statistical Machine Translation (Blatz et al. 2004; Ueffing and Ney 2007). Its basic goal is to estimate a confidence measure for each word in a given hypothesis, in order to locate those words, if any, that are likely to be incorrectly recognised or translated.

CE has been used for different purposes in Machine Translation. They have mainly been used for detecting translation errors and for improving the translation accuracy in different translation scenarios (Gandrabur and Foster, 2003; Ueffing and Ney, 2005; Blatz et al. 2003; Jayaraman and Lavie, 2005; Ueffing and Ney 2007). In this work, CE is used for detecting translation errors.

From our point of view, CE can be seen as a conventional pattern classification problem in which a feature vector is obtained for each hypothesized word in order to classify it as either correct or incorrect. Thus, our basic problems are to find appropriate pattern features and to design an accurate pattern classifier.

N-best lists have been used for different purposes in CE for speech recognition (Wessel et al. 2001) and machine translation (Blatz et al. 2004; Ueffing and Ney 2007). They have been used both to directly estimate the confidence measure and to compute predictor features. In this work, we use N-best lists to extract predictor features. Additionally, we use another feature which is based on the Model 1 proposed in (Brown et al. 1993). This feature has proved very useful in previous works (Blatz et al. 2004).

For estimating the confidence measure, we use a *smoothed naive Bayes* classification model which has been successfully used for CE in speech recognition (Sanchis et al. 2004; Sanchis et al. 2003). The model itself is a combination of *word-dependent* (specific) and *word-independent* (generalized) naive Bayes models. This classification model provides a sound framework to profitably combine the predictor features.

The paper is organized as follows. A brief review of the statistical machine translation approach is given in section 2; section 3 describes the predictor features used in this work; section 4 describes the naive Bayes classification model; section 5 presents the experimental setup, evaluation metrics and the experimental results; and, finally, section 6 contains the final conclusions.

2 Statistical Machine Translation

In the *statistical machine translation* (SMT) problem a source language word string $f_1^J = f_1 \dots f_J$ is to be translated into an optimal target language word string $\hat{e}_1^I = e_1 \dots e_I$. Such an optimal translation, is searched for among all possible target sentences of the target language, e_1^I, by applying Bayes' decision rule:

$$\begin{aligned} \hat{e}_1^I &= \arg\max_{e_1^I} P(e_1^I|f_1^J) \\ &= \arg\max_{e_1^I} \{P(e_1^I) \cdot P(f_1^J|e_1^I)\} \end{aligned} \quad (1)$$

The models adopted for each factor of Eq. 1 play an important role in SMT. On the one hand, P(e_1^I) is modeled by a language model which gives high probability to well formed target sentences. N-grams are often used for these language models. On the other hand, models for P($f_1^J|e_1^I$)

should give high probability for those sentences from the source language which are good translations for a given target sentence. These models generally consist of stochastic dictionaries, along with adequate models to account for word alignments (Brown et al. 1990; Brown et al. 1993). An alternative is to transform Eq. 1 as:

$$\hat{e}_1^I = \arg\max_{e_1^I} P(f_1^J, e_1^I) \quad (2)$$

In this case, the joint probability distribution can be adequately modeled by means of stochastic finite-state transducers (SFST) (Casacuberta and Vidal 2004) among other possible models.

3 Predictor Features

A set of features based on N-best lists has been selected to perform the experiments presented in section 5.4. These features are based on word posterior probabilities and they were proposed in (Ueffing et al. 2003).

Given an input string f_1^J in the source language, let e_i be the word which the MT system hypothesizes in the position $i \in \{1, \ldots, I\}$ of the target sentence $\hat{e}_1^I$, and let $\mathcal{L}_1^N$ be the N-best list generated by the MT system. For the computation of the features, a subset $\mathcal{S}_0^M$ of sentences is extracted from $\mathcal{L}_1^N$ based on three different criteria:

- *Levenshtein position*: $\mathcal{S}_0^M$ is composed of those M sentences containing the target word e_i in a position that is aligned to target position i in the Levenshtein alignment.
- *Target position*: $\mathcal{S}_0^M$ is composed of those M sentences containing the target word e_i in exactly the target position i.
- *Any target position*: $\mathcal{S}_0^M$ is composed of those M sentences containing the target word e_i in any position.

Different features can be calculated for e_i as:

$$\mathcal{F}(e_i) = \frac{1}{R} \sum_{\tilde{e}_1^{\tilde{I}} \in \mathcal{S}_0^M} W(\tilde{e}_1^{\tilde{I}}) \quad (3)$$

Depending on how $W(\tilde{e}_1^{\tilde{I}})$ and R are computed, three features can be defined:

- *based on sentence probabilities*: $W(\tilde{e}_1^{\tilde{I}})$ is the posterior probability of $\tilde{e}_1^{\tilde{I}}$, and R is computed by summing up the probabilities over all sentences in the N-best list.
- *based on rank weighting*: $W(\tilde{e}_1^{\tilde{I}})$ is the inverse rank of $\tilde{e}_1^{\tilde{I}}$ in the N-best list, and R is the sum of all ranks in the list.
- *based on relative frequencies*: $W(\tilde{e}_1^{\tilde{I}})$ is 1 and R is N.

Table 1: Nine features used in this work.

	Position		
	Lev.	Target	Any
Prob.	ProbLev	ProbTarget	ProbAny
Rank	RankLev	RankTarget	RankAny
Freq.	FreqLev	FreqTarget	FreqAny

Therefore, given a target word e_i, we compute 9 different features using a N-best list. We will denote these features as shown in table 1.

Additionally, we use another feature which is based on the translation Model 1 proposed by IBM in (Brown et al. 1993). Given a target word e_i, two different variants for this feature are computed: Maximal lexicon probability over all source words (Ibm1Max) and the average lexicon probability over all source words (Ibm1Av), defined as:

$$\text{Ibm1Av}(e_i) = \frac{1}{J+1} \sum_{j=0}^{J} p(e_i|f_j) \quad (4)$$

$$\text{Ibm1Max}(e_i) = \max_{0 \le j \le J} p(e_i|f_j) \quad (5)$$

where $p(e|f)$ is the lexicon probability based on IBM model 1, and f_0 is the empty source word.

4 Naive Bayes Model

We have adopted a *smoothed naive Bayes* classification model for CE. This model has been successfully used for speech recognition verification (Sanchis et al. 2004; Sanchis et al. 2003).

The class variable is denoted by c; $c = 0$ for correct and $c = 1$ for incorrect. Given a target word e and a D-dimensional vector of features $\boldsymbol{x}$, the class posteriors can be calculated via the Bayes' rule as

$$P(c|\boldsymbol{x}, e) = \frac{P(c|e)\, P(\boldsymbol{x}|c, e)}{\sum_{c'} P(c'|e)\, P(\boldsymbol{x}|c', e)} \quad (6)$$

For simplicity, the model includes the naive Bayes assumption that the features are mutually independent given a class-word pair,

$$P(\boldsymbol{x}|c, e) = \prod_{d=1}^{D} P(x_d|c, e) \quad (7)$$

Therefore, the basic problem is to estimate $P(c|e)$ for each target word and $P(\boldsymbol{x}|c, e)$ for each class-word pair. Given N training samples $\{(\boldsymbol{x}_n, c_n, e_n)\}_{n=1}^N$, the unknown probabilities can be estimated using the conventional frequencies:

$$P(c|e) = \frac{N(c, e)}{N(e)} \quad (8)$$

$$P(x_d|c, e) = \frac{N(x_d, c, e)}{N(c, e)} \quad (9)$$

where the $N(\cdot)$ are suitably defined event counts; i.e., the events are (c, e) pairs in (8) and (x_d, c, e) triplets in (9).

In practice, some features may have continuous rather than discrete domains. In that case, the use of Eq. 9 requires the discretization of continuous features. This is performed by dividing the feature domain into a fixed number of evenly-spaced bins of fixed size (usually around 20). The minimum, maximum and bin size are set by visual inspection of the histograms of the features of the examples from the correct and incorrect classes. Given this information, the naive Bayes implementation includes a function that maps the continuous feature value x_d to the corresponding discrete bin number.

Unfortunately, these frequencies often underestimate the true probabilities involving rare words and the incorrect class. To circumvent this problem, the model is smoothed using the *absolute discounting* smoothing technique imported from statistical language modelling (Ney et al. 1997). The idea is to discount a small constant $b \in (0, 1)$ to every positive count and then distribute the gained probability mass among the null counts (unseen events). A detailed explanation of the smoothed model can be found in (Sanchis et al. 2003; Sanchis et al. 2004).

Once the parameters of the model are estimated, in the test phase, a target word is classified as incorrect if the confidence estimation $P(c = 1|\boldsymbol{x}, e)$ is greater that a certain threshold τ.

5 Experimental Study

5.1 Experimental Setup

We have used the bilingual English-Spanish Xerox corpus developed in the context of the European project TransType2 (TT2 project 2002-2005). It consists of the translations of technical Xerox manuals. Basic statistics of the training, development and test sets are summarized in table 2.

Table 2: Statistics of the English-Spanish Xerox corpus.

		English	Spanish
Train	Sentences	55.761	
	Running words	665.400	750.691
	without PM*	568.746	652.124
	Vocabulary size	7.956	10.622
Dev.	Sentences	1.012	
	Running words	14.278	15.574
	without PM	11.903	13.260
	Vocabulary size	1.224	1.409
Test	Sentences	1.125	
	Running words	8.370	9.551
	without PM	7.559	8.762
	Vocabulary size	1.132	1.164

(*) PM: Punctuation Marks

Using the GIATI statistical finite-state transducer approach (Casacuberta and Vidal 2004; Civera et al. 2004) approximately 10.000-best lists were generated for each source sentence in order to extract the predictor features presented in section 3.

5.2 Confidence Tagging

In order to evaluate the performance of the predictor features and the classification model described in sections 3 and 4, respectively, a corpora is needed where each automatically translated word is tagged as correct or incorrect.

Automatically tagging the translated words as correct or incorrect can be done by comparing the translation to several references, though in this work we have only used one reference. We have considered three different tagging methods.

1. Word Error Rate (WER): Each hypothesized word is tagged as correct if it is Levenshtein-aligned to itself in the reference.

2. Position-independent Error Rate (PER): Each hypothesized word is searched in the whole reference and, if found, it is drawn *without* replacement and tagged as correct.

3. Position-independent Error Rate with Replacement (PERR): Each hypothesized word is searched in the whole reference and, if found, it is drawn *with* replacement and tagged as correct.

From these definitions, it is clear that $WER \geq PER \geq PERR$.

5.3 Evaluation Metrics

Given a certain translation task, let us assume that using a tagging method we obtain N_c words tagged as correct and N_i words tagged as incorrect. Then, after confidence classification is performed for a certain classification threshold τ, let us assume that we obtain $0 \leq N_f(\tau) \leq N_c$ words tagged as correct which are classified as incorrect (false rejection), and $0 \leq N_t(\tau) \leq N_i$ tagged as incorrect which are classified as incorrect (true rejection).

Based on the false rejection $N_f(\tau)$ and the true rejection $N_t(\tau)$, two measures are of interest for the evaluation of CE:

1. The *False Rejection Rate*, defined as:

$$R_f(\tau) = \frac{N_f(\tau)}{N_c} \tag{10}$$

2. The *True Rejection Rate*, defined as:

$$R_t(\tau) = \frac{N_t(\tau)}{N_i} \tag{11}$$

The trade-off between R_f and R_t values depends on the decision threshold τ. A *Receiver Operating Characteristic* (ROC) curve represents R_f against R_t for different values of $\tau \in [0, 1]$.

The area under a ROC curve divided by the area of a worst-case diagonal ROC curve, provides an adequate overall estimation of the classification accuracy. We denote this area ratio as AROC. The AROC value is in the range of 1.0 to 2.0. Note that an AROC value of 2.0 would indicate that all words can be correctly classified.

Another different criterion is the *Confidence Error Rate* (CER). This metric is defined as the number of classification errors divided by the total number of classified words. Thus, the CER value also depends on the decision threshold τ. CER can be computed as:

$$CER(\tau) = \frac{N_f(\tau) + (N_i - N_t(\tau))}{N_c + N_i} \quad (12)$$

A baseline CER is obtained assuming that all target words are classified as correct. Then, the baseline CER is:

$$CER_{baseline} = \frac{N_i}{N_c + N_i} \quad (13)$$

5.4 Experimental Results

The unknown probabilities of the *smoothed naive Bayes* model, presented in section 4, were estimated using the training set. Different smooth parameters of the model were optimized using the development set. Also, the development set was used to find the best classification threshold τ i.e., that with minimum CER(τ).

The test set was classified in different manners. First of all, in order to evaluate the performance of each single feature, we classified the test set using the smoothed model based on one-dimensional feature vectors. To further exploit the usefulness of the features, the naive Bayes model was employed to explore the performance of a large number of different feature combinations.

The results achieved are shown in table 3 for WER, PER, and PERR tagging methods. Based on the single feature performance, we can divide the features into four main groups. The first group is only composed by the best single feature Ibm1Max. Although this feature does not obtain consistently the best AROC values, it achieves the most significant relative reductions on baseline CER: 16.6%, 17.9% and 20% for WER, PER and PERR, respectively. The second group is composed of the n-best list based features computed using target positions. This criterion achieves the best performance. The third group is composed of the other two features based on sentence probabilities: ProbLev and ProbAny. These two features achieve similar CER values than the second group of features, but lower AROC values. It seems that the use of sentence probabilities helps to reduce the negative effect of the position criteria in the computation of these two features. The last group is composed by the features which, in general, get the lower AROC and CER values. These group show that the use of the ranking and relative frequencies, along with the Levenshtein and any position criteria, does not achieve good performance. The feature Ibm1Av is clearly the worst feature for CER. This is surprising since the Ibm1Max is the best one. A possible explanation is that averaging reduce significantly the lexicon probability. For this reason, all the target words get low values for this feature. The discretization of the feature does not produce a good class predictor.

Through the (naive Bayes) combination of the best single feature Ibm1Max along with ProbLev, the classification accuracy is improved for the PER and PERR tagging methods. This feature combination achieves the higher AROC and CER values, with a relative reductions of baseline CER of 20.8% and 22.5%, for PER and PERR, respectively.

Figures 1, 2 and 3, show the comparative test set ROC curves, for the WER, PER and PERR tagging methods, respectively.

6 Final Remarks

Confidence estimation can be considered as a classical pattern classification problem in two possible classes: correct or incorrect. Thus, our basic problems are to find appropriate pattern features and to design an accurate pattern classifier.

As pattern features, we used a set of n-best list based features along with features based on the Model 1 proposed by IBM. As classifier, we used a naive Bayes model which provides a sound framework to profitably combine the predictor features.

Experiments were performed using a bilingual English-Spanish corpus which contains the translations of technical manuals.

The results presented confirm those of previous works (Blatz et al. 2004) showing that features based on IBM Model 1 are useful to detect incorrectly translated words. The n-best list based features which are computed using posterior probabilities achieve best performance than those based on relative frequencies or rank weights. Also, the consideration of target positions in the computation of these features appears as the best criterion.

The *naive Bayes* feature combination produces better classification accuracy than the single feature performance. However, we have achieved important relative reduction in baseline CER for both single and combined feature performance.

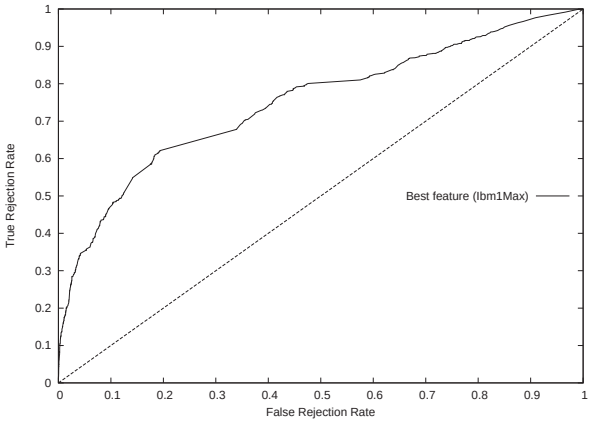

Figure 1: ROC curve on the test set for the best feature (WER).

Acknowledgements

Work supported by the EC (FEDER) and the Spanish MEC under grant TIN2006-15694-CO2-01.

Table 3: AROC and CER [%] for each single feature and the best combination. The best CER and AROC values for each tagging method are in boldface.

	WER		PER		PERR	
Features	AROC	CER	AROC	CER	AROC	CER
ProbLev+Ibm1Max	1.57	17.4	**1.67**	**13.7**	**1.73**	**11.7**
Ibm1Max	1.50	**17.1**	1.59	14.2	1.67	12.1
ProbTarget	**1.66**	18.4	1.66	16.3	1.68	13.9
RankTarget	1.57	18.8	1.60	15.9	1.62	14.0
FreqTarget	1.55	19.1	1.59	16.2	1.61	14.2
ProbLev	1.43	19.0	1.45	15.9	1.51	13.3
ProbAny	1.41	18.4	1.45	16.9	1.52	14.1
RankAny	1.34	19.7	1.40	16.6	1.46	14.4
RankLev	1.30	19.9	1.36	16.9	1.42	14.7
FreqLev	1.30	20.4	1.36	17.1	1.43	15.0
FreqAny	1.34	20.4	1.38	17.2	1.45	15.0
Ibm1Av	1.47	20.4	1.50	17.3	1.60	15.1
Baseline	–	20.5	–	17.3	–	15.1

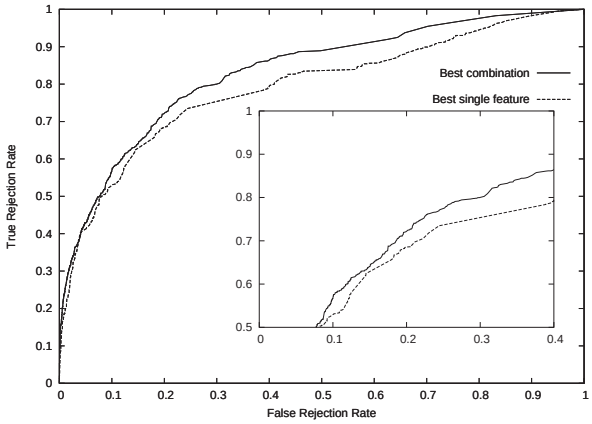

Figure 2: ROC curve on the test set for the best single feature and the best combination (PER).

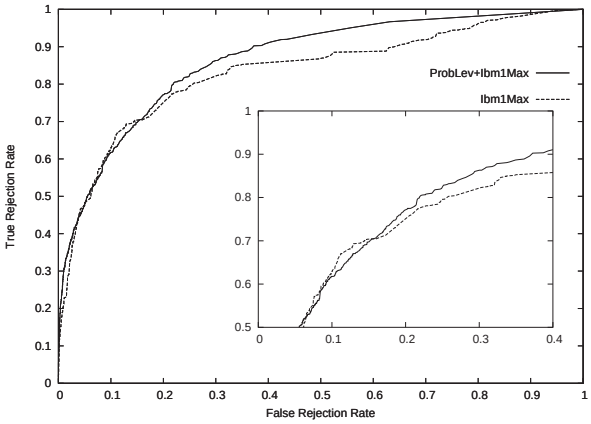

Figure 3: ROC curve on the test set for the best single feature and the best combination (PERR).

References

J. Blatz, E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis and N. Ueffing. Confidence estimation for machine translation. Final report, JHU/CLSP SummerWorkshop.

J. Blatz, E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis, and N. Ueffing. Confidence estimation for machine translation. In Proc. of *COLING'04*, pages 315-321, Geneva, 2004.

P.F. Brown, J. Cocke, S.A. Della Pietra, V.J. Della Pietra, F. Jelinek, J.D. Lafferty, R.L. Mercer, and P.S. Roossin. A Statistical Approach to Machine Translation. *Computational Linguistics*, 16(2):79–85, 1990.

P.F. Brown, S.A. Della Pietra, V.J. Della Pietra, and R.L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, 1993.

F. Casacuberta and E. Vidal. Machine translation with inferred stochastic finite-state transducers. *Computational Linguistics*, 30(2):205–225, 2004.

J. Civera, J. M. Vilar, E. Cubel, A. L. Lagarda, S. Barrachina, E. Vidal, F. Casacuberta, D. Picó, and J. González. From machine translation to computer assisted translation using finite-state models. In Proc. of the 2004 *Conference on Empirical Methods in Natural Language Processing (EMNLP04)*, Barcelona, 2004.

S. Gandrabur and G. Foster. Confidence estimation for text prediction. In Proc. of *Conference on Natural Language Learning (CoNLL)*, pages 95–102, 2003.

S. Jayaraman and A. Lavie. Multi-engine machine translation guided by explicit word matching. In Proc. of the *10th Annual Conference of the European Association for Machine Translation (EAMT)*, pages 143–152, 2005.

H. Ney, S. Martin and F. Wessel. Statistical language modeling using leaving-one-out. *Young and Bloothoft, editors, Corpus Based Methods in Language and Speech Processing*, pp. 174–207, 1997.

A. Sanchis, A. Juan, and E. Vidal. Improving utterance verification using a smoothed naive bayes model. In Proc. of the *IEEE ICASSP'2003*, vol. 1, pp. 592–595, 2003.

A. Sanchis. *Estimación y aplicación de medidas de confianza en reconocimiento automático del habla*. PhD thesis. 2004.

N. Ueffing, K. Macherey, and H. Ney. Confidence measures for statistical machine translation. In Proc. of *MT Summit IX*, pages 394-401, 2003.

N. Ueffing and H. Ney. Application of word-level confidence measures in interactive statistical machine translation. In Proc. of the *10th Annual Conference of the European Association for Machine Translation (EAMT)*, pages 262–270, 2005.

N. Ueffing and H. Ney. Word-level confidence estimation for machine translation. Computational Linguistics, March 2007, 33(1):9–40.

F. Wessel, R.Schlüter, K.Macherey and H.Ney. Confidence measures for large vocabulary continuous speech recognition. *IEEE Trans. on Speech and Audio Processing*, 9(3):288–298, 2001.

TransType2 - Computer Assisted Translation. RTD project TransType2 (IST-2001-32091) funded by the European Commission. http://tt2.sema.es