Partial Least Squares for Word Confidence Estimation in Machine Translation

Jesús González-Rubio[1], J.Ramón Navarro-Cerdán[2], and Francisco Casacuberta[1]

[1] Universitat Politècnica de València, Camino de Vera s/n, 46022, Valencia, Spain, {jegonzalez,fcn}@dsic.upv.es
[2] Instituto Tecnológico de Informática, Camino de Vera s/n, 46022,Valencia, Spain, jonacer@iti.upv.es

Abstract We present a new technique to estimate the reliability of the words in automatically generated translations. Our approach addresses confidence estimation as a classification problem where a confidence score is to be predicted from a feature vector that represents each translated word. We describe a new set of prediction features designed to capture context information, and propose a model based on partial least squares to perform the classification. Good empirical results are reported in a large-domain news translation task.

Keywords: Machine translation, Word confidence estimation, Statistical multivariate analysis, Partial least squares discriminant analysis

1 Introduction

Despite an intensive research in the last twenty years, the pattern recognition approach to translation, known as statistical machine translation (SMT), is still far from perfect [1]. Thus, a desirable feature to improve its broader and more effective deployment is the capability of predicting the reliability of the generated translations. This task is referred to as confidence estimation (CE).

Following previous woks in the literature [2,3], we address CE for translated words as a conventional pattern classification problem in which a feature vector is obtained for each word in order to classify it as either correct or incorrect. This point of view provides a solid, well-know framework, within which accurate two-class classifiers can be derived. The challenges of this approach are to find an appropriate set of features, and to learn accurate classification models.

Ueffing et al. [2] were the first to apply word posterior probabilities, very effective in speech recognition, to estimate MT confidences. They compute such probabilities from N-best lists of translations, and use them as direct estimations of the reliability of the translated words. Sanchís et al. [3] proposed new approaches to compute features from N-best lists, similarly as posterior probabilities are computed in [2]. Moreover, they proposed a smoothed naïve Bayes classifier to combine these features and improve prediction accuracy. Since naïve

Bayes models work on discrete domains, continuous features such as these must be mapped to a discrete domain which involves an additional tuning step.

Our work extends previous approaches in several aspects, including the addition of new features, and the use of a novel classification model based on multidimensional statistical analysis. As [3], we also compute prediction features based on posterior probabilities. However, we generalize this approach to take into account the context of each word. The key idea is that the reliability of a word is influenced by the context in which it appears, therefore by using context-aware features we expect to obtain a stronger estimation of each word reliability. Additionally, we propose a new classifier based on the partial least squares [4]. This classifier performs an intrinsic transformation of the features such that a maximum separation among classes is obtained. Thus it is an effective and efficient method that allows us to build robust classifiers even for ambiguous and redundant features such the ones in natural language processing.

The rest of the article is organized as follows. A brief review of SMT is given in Section 2; Section 3 describes the predictor features used in the experimentation; Section 4 describes the classifier which is based on partial least squares discriminant analysis; Section 5 presents the experimental setup, the assessment measures, and the results of the experiments; and, finally, Section 6 provides a summary and presents the final conclusions.

2 Statistical Machine Translation

SMT formalizes the translation problem as follows. Given a source language sentence $\boldsymbol{s} \in \mathcal{S}$, the goal is to obtain its equivalent target language translation $\boldsymbol{t} \in \mathcal{T}$. From the set of all possible target language sentences, we are interested in that $\hat{\boldsymbol{t}}$ with the highest probability[3]:

$$\hat{\boldsymbol{t}} = \arg\max_{\boldsymbol{t} \in \mathcal{T}} Pr(\boldsymbol{t}|\boldsymbol{s}) \tag{1}$$

The posterior probability $Pr(\boldsymbol{t}|\boldsymbol{s})$ is usually modeled by a log-linear, namely a maximum-entropy [5], model. The posterior probability is computed from a set of feature functions $f_m(\boldsymbol{t}, \boldsymbol{s})$ and a corresponding set of weights λ_m:

$$Pr(\boldsymbol{t}|\boldsymbol{s}) \approx P_{\overrightarrow{\lambda}}(\boldsymbol{t}|\boldsymbol{s}) = \frac{\exp\left(\sum_m \lambda_m f_m(\boldsymbol{t}, \boldsymbol{s})\right)}{\sum_{\boldsymbol{t}' \in \mathcal{T}} \exp\left(\sum_m \lambda_m f_m(\boldsymbol{t}', \boldsymbol{s})\right)} \tag{2}$$

Since the normalization term in the denominator does not depend on the target sentence $\boldsymbol{t}$, it can be omitted during the search process. Thus, the optimum target language sentence $\hat{\boldsymbol{t}}$ can be finally computed as:

$$\hat{\boldsymbol{t}} = \arg\max_{\boldsymbol{t} \in \mathcal{T}} \sum_m \lambda_m f_m(\boldsymbol{t}|\boldsymbol{s}) \tag{3}$$

[3] We use $Pr(\cdot)$ to denote general probability distributions, and $P(\cdot)$ to denote model-based distributions.

3 Prediction Features

Now, we describe the set of prediction features used in the experimentation. As in [3], our features are based on posterior probabilities computed from N-best lists [2]. However, we generalize this approach to take into account the context of each target word. As done in language modeling, we consider the preceding words as the context, namely the history, of each target word.

Given a source sentence $\boldsymbol{s}$, let $\boldsymbol{t} = t_1 \dots t_i \dots t_{|\boldsymbol{t}|}$ be the translation hypothesized by an SMT system, and let $\mathcal{L} = \{\boldsymbol{t}_1, \dots \boldsymbol{t}_n, \dots, \boldsymbol{t}_N\}$ be the corresponding N-best list of translations. $\mathcal{L}$ is ordered by a given score $W(\boldsymbol{t}_n)$ assigned to each translation. For each word $t_i \in \boldsymbol{t}$ with history $\tilde{\boldsymbol{h}}_c(t_i) = t_{i-c} \dots t_{i-1}$ of size c, we compute features $F(t_i, c)$ by summing up the scores of those translations in $\mathcal{L}$ that contain word t_i with history $\tilde{\boldsymbol{h}}_c(t_i)$ in a position aligned to position i of $\boldsymbol{t}$:

$$F(t_i, c) = \frac{1}{Z} \sum_{\substack{\boldsymbol{t}' \in \mathcal{L}: \\ t'_{A(\boldsymbol{t}', t_i)} = t_i \\ \tilde{\boldsymbol{h}}_c(t'_{A(\boldsymbol{t}', t_i)}) = \tilde{\boldsymbol{h}}_c(t_i)}} W(\boldsymbol{t}') \tag{4}$$

where $A(\boldsymbol{t}', t_i)$ is an alignment function that returns the position in $\boldsymbol{t}'$ aligned to word t_i. Therefore, $t'_{A(\boldsymbol{t}', t_i)}$ is the actual word in $\boldsymbol{t}'$ aligned to t_i, and $\tilde{\boldsymbol{h}}_c(t'_{A(\boldsymbol{t}', t_i)})$ is its history. We sum the scores of those sentences for which the aligned word $t'_{A(\boldsymbol{t}', t_i)}$ is equal to t_i and the history of the aligned word $\tilde{\boldsymbol{h}}_c(t'_{A(\boldsymbol{t}', t_i)})$ is equal to the history of t_i. Finally, $Z = \sum_{\boldsymbol{t}' \in \mathcal{L}} W(\boldsymbol{t}')$ is a normalization term introduced to obtain probability-like features.

The actual value of the feature $F(t_i, c)$ in Equation (4) depends on the definition of the alignment $A(\boldsymbol{t}', t_i)$ and score $W(\boldsymbol{t}')$ functions. Following [3], we consider three different alignment methods:

- Lev: Levensthein alignment [6]
- Target: Word t_i is aligned to position i in $\boldsymbol{t}'$
- Any: Word t_i is aligned to any position i' in $\boldsymbol{t}'$ so that $t'_{i'} = t_i$

and three different scoring schemes:

- Prob: The score of $\boldsymbol{t}'$ is the probability $P(\boldsymbol{t}'|\boldsymbol{s})$ given by the SMT model
- Rank: The score of $\boldsymbol{t}'$ is one divided by the position of $\boldsymbol{t}'$ in $\mathcal{L}$
- Freq: All translations are assigned equal score: $\forall \boldsymbol{t}' \in \mathcal{L}, W(\boldsymbol{t}') = 1.0$

Thus, given a target word t_i, we compute nine different features for each history size c. We name each feature with the size of the context, the scoring scheme, and the alignment method. E.g., 0-ProvLev stands for history size equal to zero, probability scoring, and Levensthein alignment. Note that the nine features for a history size $c = 0$ are equal to those described in [3].

We compute two additional features based on the simple SMT model 1 [7] by IBM. Again, let $\boldsymbol{t}$ be the translation of source sentence $\boldsymbol{s} = s_1 \dots s_j \dots s_{|\boldsymbol{s}|}$. The features are given by the average lexicon probability of word t_i over all source words (Ibm1Avg), and its maximal lexicon probability (Ibm1Max):

$$\text{Ibm1Avg}(t_i) = \frac{1}{|\boldsymbol{s}| + 1} \sum_{j=0}^{|\boldsymbol{s}|} P(t_i|s_j) \qquad \text{Ibm1Max}(t_i) = \max_{0 \le j \le |\boldsymbol{s}|} P(t_i|s_j) \tag{5}$$

where $P(t_i|s_j)$ is the model 1 probability, and s_0 is the empty source word.

We consider history sizes from zero to three. Therefore, we compute a total of 36 N-best features and two additional model-1-based features, for a total of 38 features. Additionally, note that the set of 11 features described in [3] is a subset of our 38 features: the nine N-best features with history size $c = 0$ plus the two model-1-based features.

4 Partial Least Squares Discriminant Analysis

We formalize CE as a classification problem $\mathbb{R}^m \rightarrow \{0, 1\}$ where we predict a discrete variable $y \in \{0, 1\}$ (0 denotes an incorrect word and 1 a correct one) given a vector of m explanatory variables $\boldsymbol{x}^T = (x_1, \ldots, x_m)$ (the set of features that represents each word). The features described in the previous section are obviously highly redundant, thus the selected learning method has to be robust in the presence of noisy data. We chose to use partial least squares discriminant analysis [4] (PLS-DA). PLS-DA performs an implicit transformation of a set of possibly noisy features into a set of uncorrelated latent variables that still account for the maximum co-variability between the features and the discrete variable y. Formally, given a training set $\{\boldsymbol{x}_i, y_i\}_{i=0}^n$, let $\boldsymbol{X}$ be a matrix where each row is the feature vector $\boldsymbol{x}_i$ of a training sample, and $\boldsymbol{y}$ a vector with the class y_i of each sample, PLS-DA builds the following linear model:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{b} + \boldsymbol{f} \tag{6}$$

The estimation of the regression coefficients $\boldsymbol{b}$ for PLS-DA is different from the conventional least squares regression. The intuitive idea of PLS-DA is to describe $\boldsymbol{y}$ as well as possible, hence to make the vector or Gaussian errors $||\boldsymbol{f}||$ as small as possible, and, at the same time, take advantage of the relation between $\boldsymbol{X}$ and $\boldsymbol{y}$. To do that, PLS-DA defines two independent transformations $\boldsymbol{P}$ and $\boldsymbol{q}$ (for $\boldsymbol{X}$ and $\boldsymbol{y}$ respectively) with $\boldsymbol{E}$ and $\boldsymbol{f}$ being the corresponding residual errors, and a linear relation $\boldsymbol{R}$ linking both blocks:

$$\boldsymbol{X} = \boldsymbol{T}\boldsymbol{P}^T + \boldsymbol{E} \qquad \boldsymbol{y} = \boldsymbol{U}\boldsymbol{q}^T + \boldsymbol{f} \tag{7}$$

$$\boldsymbol{U} = \boldsymbol{T}\boldsymbol{R} \tag{8}$$

where matrices $\boldsymbol{T}$ and $\boldsymbol{U}$ are the projections from $\boldsymbol{X}$ and $\boldsymbol{y}$ respectively. Specifically, each of the columns of the $\boldsymbol{T}$ matrix represents one of the latent variables of $\boldsymbol{X}$. The NIPALS algorithm [4] is used to solve this optimization problem. In this case, $\boldsymbol{b}$ is estimated as:

$$\boldsymbol{b} = \boldsymbol{R}\boldsymbol{q}^T \quad \text{where} \quad \boldsymbol{R} = \boldsymbol{W}(\boldsymbol{P}^T\boldsymbol{W})^{-1} \tag{9}$$

where $\boldsymbol{W}$ is an internal weight matrix used by the algorithm that accounts for the correlation between $\boldsymbol{X}$ and $\boldsymbol{U}$. The predictions of the PLS-DA model are in the range $[0, 1]$, thus, each word is finally classified as either correct or incorrect depending on whether its score exceeds or not a given classification threshold τ.

In the experiments, we use the PLS-DA classifier implemented by the `pls` library of the R toolkit [8]. The dimension of the intrinsic reduction performed by PLS-DA remains as the only free parameter of the classifier.

5 Experiments

5.1 Experimental Setup

We computed word confidence scores for translations of the English-Spanish news evaluation data used in the quality estimation task of the 2012 workshop on statistical machine translation [9]. Those translations were generated by a log-linear SMT model trained on the Europarl and News Commentaries corpora as provided in the same workshop for the shared translation task. We used this same training data to build the Model 1 model required for lbm1Avg and lbm1Max features. Evaluation data contains 1832 translations for training and 422 translations for test. Additionally, for each translation the corresponding source sentence, and list of 1000-best translation options are also available. We use these to compute features for every word in the lowecased and tokenized version of the corpora. This process results in a training corpus with ~ 55 thousand samples and a test corpus with ~ 11 thousand samples, each sample being a 38-dimensional float vector of features.

The optimum classification threshold τ, and the optimum values for the free parameters of the models were estimated by ten-fold cross validation on the training corpus. As a baseline, we compare the performance of the PLS-DA classifier to the smoothed naïve Bayes model proposed in [3].

5.2 Confidence Tagging

To evaluate the performance of our confidence estimations, each word has to be tagged as either correct or incorrect. Since manual tagging of the words by human experts is a slow and expensive task, we automatically tag translated words by comparing each translation to the reference translation in the corpus. We follow [3] and consider three different tagging methods[4]:

Word error rate (WER): Each translated word is tagged as correct if it is Levensthein-aligned [6] to itself in the reference.

Position-independent error rate (PER): The word is searched in the reference, and, if found, it is drawn *without* replacement and tagged as correct.

Position-independent error rate with replacement (PERR): The word is searched in the reference, and, if found, it is drawn *with* replacement and tagged as correct.

5.3 Assessment Measures

Let us assume a translation task that contains N_c words tagged as correct and N_i words tagged as incorrect. Then after confidence classification is performed for a certain threshold value, let us assume that $\mathrm{FP}(\tau)$ of the words tagged as incorrect are classified as correct $0 \leq \mathrm{FP}(\tau) \leq N_i$ (false positives), and $0 \leq \mathrm{TP}(\tau) \leq N_c$ words tagged as correct that are classified as correct (true positives). Then, we

[4] Note that WER is the most strict tagging while PERR is the most tolerant.

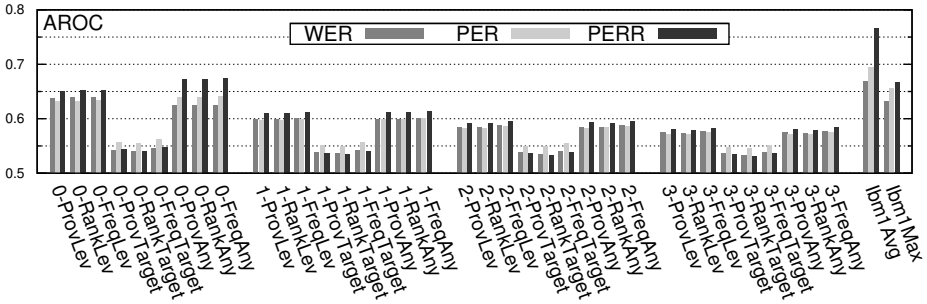


Figure 1. Cross-validation AROC values of each feature when used as a direct estimator of the reliability of the translated words in the training set.

define the false positive rate $\mathrm{FPR}(\tau)$, the true positive rate $\mathrm{TPR}(\tau)$ and the classification accuracy, namely the confidence error rate $\mathrm{CER}(\tau)$:

$$\mathrm{FPR}(\tau) = \frac{\mathrm{FP}(\tau)}{N_i} \quad \mathrm{TPR}(\tau) = \frac{\mathrm{TP}(\tau)}{N_c} \qquad \mathrm{CER}(\tau) = \frac{\mathrm{FP}(\tau) + (N_c - \mathrm{TP}(\tau))}{N_c + N_i}$$

The trade-off between $\mathrm{FPR}(\tau)$ and $\mathrm{TPR}(\tau)$ can be represented in a so called receiver operating characteristic (ROC) curve. ROC curves are calculated by using different threshold values $\tau \in [0, 1]$ to classify the words and keeping track of $\mathrm{FPR}(\tau)$ and $\mathrm{TPR}(\tau)$. The area under the ROC curve (AROC) provides an adequate overall estimation of the classification accuracy.

We also compute the statistical significance of the observed performance differences. Specifically, we compute p-values using a randomization version of the paired t-test [10]. First, we use a evaluation measure, e.g. AROC, to determine the performance difference between the outcomes of two methods. Then, we repeatedly create shuffled versions of the original outcomes, determine the performance difference between them, and count the number of times that this difference is equal or larger than the original difference. Finally, the p-value is the proportion of iterations in which the difference was indeed larger for the shuffled version.

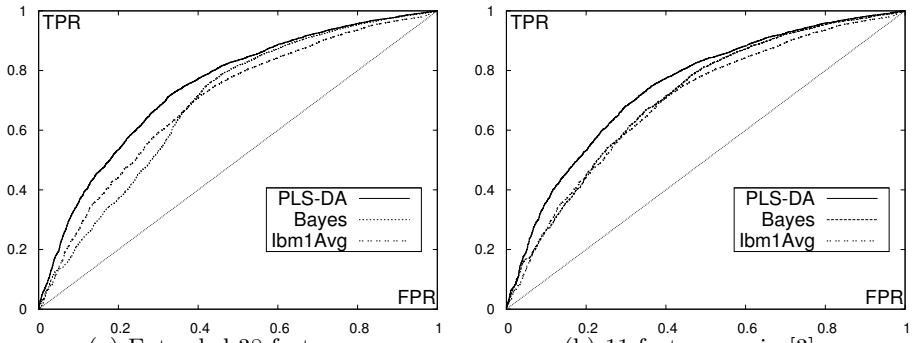
5.4 Results

In a first experiment, we studied the performance of each individual feature described in Section 3 as a direct estimation of the quality of the words. Figure 1 displays the cross-validation AROC obtained by each feature for the three tagging criteria. Results show that the model-1-based features obtained the best performance. Regarding posterior probability features, Lev and Any alignment methods consistently outperformed Target alignment for all history sizes. Also, features obtained slightly worse results as larger history sizes were used.

Then, we evaluated the classification accuracy of the PLS-DA model presented in Section 4 in comparison to the naïve Bayes classifier proposed in [3], and the best-performing feature Ibm1Avg. We present results for two sets of features, the 11 features used in [3], and the extended set of 38 features proposed here. Table 1 displays the AROC and CER scores obtained in the test set.

Feature set	Method	WER		PER		PERR	
		AROC	CER	AROC	CER	AROC	CER
Ibm1Avg (best individual feature)		0.70	0.33	0.72	0.27	0.77	0.24
11 features as in [3]	naïve Bayes	0.70	0.29	0.75	0.25	0.76	0.22
	PLS-DA	**0.75****	**0.28***	**0.76****	**0.24**	**0.82****	**0.21****
Extended 38 features	naïve Bayes	0.69	0.29	0.70	0.25	0.74	0.23
	PLS-DA	**0.75****	**0.28***	**0.76****	**0.24***	**0.81****	**0.21****

Table 1. AROC and CER results in the test set. Best results are displayed in bold. We use asterisks ($*$) to denote the statistical significance of the AROC and CER differences observed between PLS-DA and naïve Bayes: $^{*}p < 0.01$, $^{**}p < 0.001$.



(a) Extended 38 features (b) 11 features as in [3]

Figure 2. ROC curves for results in the test set using WER tagging.

Results show that the proposed PLS-DA model consistently outperformed the baseline naïve Bayes model for all tagging criteria and feature sets. Additionally, results for naïve Bayes clearly deteriorated, e.g. -0.05 AROC for PER tagging, when the extended feature set was used. In fact, naïve Bayes obtained worse AROC results with the extended feature set than the single best-performing feature. This fact indicates that the naïve Bayes model had difficulties in handling highly-correlated features of different quality, and these learning issues get worse as more features are used. I.e., to obtain a good performance, it would require a previous feature selection step to filter out redundancy and low-quality features. In contrast, PLS-DA not only obtained better performance for the "high-quality" 11-feature set in [3], but its performance did not degraded for the highly-redundant extended set of features. These results indicate that PLS-DA is a better-performing and more robust classification model than the baseline naïve Bayes model. Finally, Figure 2 shows for the WER tagging method the test set ROC curves for PLS-DA, naïve Bayes, and the best individual feature.

6 Summary and Future Work

We have presented a new CE method to classify as correct or incorrect the words of the translations generated by an SMT system. For each word, we compute a number of features, and use a partial least squares model to classify the word.

Our results showed that features based on the model 1 lexicon achieve the better performance, followed by those computed from N-best with the Lev and Any alignment criteria. Also, the use of context information did not improve the results of the previously used [2,3] N-best-based features. Regarding the classification models, the proposed PLS-DA model consistently outperformed the smoothed naïve Bayes model proposed in [3] in all test conditions. In fact, PLS-DA had shown to be an effective, scalable, and robust classification model quite adequate for the task

As future work, we plan to exploit the scalability of the PLS-DA model to study interactions between the features. Additionally, we will explore techniques to automatically estimate the optimum number of latent variables to use.

Acknowledgements

We thank Alberto Sanchis for providing us with the code of the naïve Bayes model. Work supported by the European Union Seventh Framework Program (FP7/2007-2013) under the CasMaCat project (grans agreement nº 287576), by Spanish MICINN under TIASA (TIN2009-14205-C04-02) project, and by the Generalitat Valenciana under grant ALMPR (Prometeo/2009/014).

References

1. NIST: National Institute of Standards and Technology MT evaluation official results. http://www.itl.nist.gov/iad/mig/tests/mt/ (November 2006)
2. Ueffing, N., Macherey, K., Ney, H.: Confidence measures for statistical machine translation. In: Proc. of the MT Summit, Springer-Verlag (2003) 394–401
3. Sanchis, A., Juan, A., Vidal, E.: Estimation of confidence measures for machine translation. In: Proc. of the Machine Translation Summit. (2007) 407–412
4. Wold, H. In: Estimation of Principal Components and Related Models by Iterative Least squares. Academic Press, New York (1966) 391–420
5. Berger, A.L., Pietra, V.J.D., Pietra, S.A.D.: A maximum entropy approach to natural language processing. Computational Linguistics **22** (March 1996) 39–71
6. Levenshtein, V.: Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics Doklady **10**(8) (February 1966) 707–710 Originally appeared as: В.И. Левенштейн (1965). "Двоичные коды с исправлением выпадений, вставок и замещений символов". Доклады Академий Наук СССР 163 (4): 845–8.
7. Brown, P., Della Pietra, V., Della Pietra, S., Mercer, R.: The mathematics of statistical machine translation: parameter estimation. Computational Linguistics **19** (1993) 263–311
8. Mevik, B.H., Wehrens, R., Liland, K.H.: pls: Partial Least Squares and Principal Component regression. (2011) R package version 2.3-0.
9. Callison-Burch, C., Koehn, P., Monz, C., Post, M., Soricut, R., Specia, L.: Findings of the 2012 workshop on statistical machine translation. In: Proc. of the Workshop on Statistical Machine Translation, Montréal, Canada (June 2012) 10–51
10. Chinchor, N.: The statistical significance of the muc-4 results. In: Proceedings of the Conference on Message Understanding. (1992) 30–50