MT Journal manuscript No.
(will be inserted by the editor)

Sentence-level ranking with quality estimation

Eleftherios Avramidis

Received: date / Accepted: date

Abstract Starting from human annotations, we provide a strategy based on machine learning that performs preference ranking on alternative machine translations of the same source, at sentence level. Rankings are decomposed into pairwise comparisons so that they can be learned by binary classifiers, using black-box features derived from linguistic analysis. In order to recompose from the pairwise decisions of the classifier, they are weighed with their classification probabilities, increasing the correlation coefficient by 80%. We also demonstrate several configurations of successful automatic ranking models. The best configurations achieve a correlation with human judgments measured by Kendall's tau at 0.27. Although the method does not use reference translations, this correlation is comparable to the one achieved by state-of-the-art reference-aware automatic evaluation metrics such as smoothed BLEU, METEOR and Levenshtein distance.

Keywords quality estimation · ranking · logistic regression · linguistic features · sentence selection

1 Introduction

The ongoing integration of Machine Translation (MT) in professional workflows has increased the need for an estimation of quality of the produced output. This need has defined a new area of research, which combines machine-learning techniques and linguistic observations into a post-processing mechanism able to provide useful hints about the translation success.

Eleftherios Avramidis
German Research Center for Artificial Intelligence (DFKI GmbH)
Language Technology Lab
Alt Moabit 91c, Berlin, Germany
Tel.: +49-30 238 95-1806
Fax: +49-30 238 95-1810
E-mail: eleftherios.avramidis@dfki.de

Whereas research on *Quality Estimation* has recently focused on predicting absolute single-sentence quality scores, there have been concerns on how confident one can be in quantifying quality; particularly in defining the distinction between the level of quality that each score represents (Callison-Burch et al, 2007). We therefore attempt to look at the problem from a different perspective. Our goal is to achieve *comparative* quality estimation in the sense of comparing the output of several translation systems with each other, given the same input, but lacking reference translations. The knowledge for the task is obtained by machine learning on existing human judgements.

Consequently, we present a method of *machine ranking* which performs automatic quality ranking of several alternative translation outputs of the same source sentence. As human rankings have been long used as means for MT Evaluation, they serve as a training material for building a machine learning approach that can reproduce ranking over multiple translation outputs. Several fluency and adequacy criteria are used to feed the machine learning process. The aspects that this work investigates are:

- the ability to learn from human rankings how to compare alternative translations of the same sentence
- the approach of using pairwise classification confidence in order to reconstruct unambiguous rankings
- the comparison of its ranking ability with that of reference-aware metrics
- the appropriateness of using it for system combination through sentence selection

After a short review of previous related work (section 2), we define the problem and describe the methods (section 3), including the underlying pairwise mechanism, the machine learning algorithms, the features used and the evaluation. Section 4 includes the setup of the experiment and the strategy that was followed to select the best systems and evaluate them, as well as the findings of the results. The final conclusions are given in section 5.

2 Related work

The field of *Quality Estimation* tries to provide quality assessment on the translation output without access to reference translations. Relevant research includes statistical methods on predicting word-level confidence (Ueffing and Ney, 2005; Raybaud et al, 2009b) or correctness of a sentence (Blatz et al, 2004), whereas the problem has been also seen as a regression task for estimating correctness scores or probabilities (Specia et al, 2009; Raybaud et al, 2009a). The focus of these works is on estimating absolute measures of quality.

Closer to our goal on comparative estimation of quality among several system outputs, there have been a few approaches, most of them adapted to particular applications. Concerning **System Combination**, Rosti et al (2007) perform sentence-level selection by using generalised linear models, based on re-ranking N-best lists merged from many MT systems. Sánchez-Martínez (2011) builds a classifier which decides the best machine translation system to translate a sentence by using only source-language information, but shows a small, non-significant improvement. Specia et al (2010) build one Quality Estimation model for each MT system and then use the scores from these individual models on each sentence to rank alternative translations of the same source. Other approaches (Vilar et al, 2011; Soricut and Narsale, 2012) use machine learning for ranking the candidate translations and then selecting the highest-ranked translation as the final output. He et al (2010) train a binary classifier for sentence selection between two outputs, originating from a statistical MT system and a Translation Memory. In Avramidis (2011), a sentence-selection approach is trained based on Levenshtein distance with the reference translations and internal translation features, performing better than a state-of-the-art system-combination system (Federmann et al, 2012). **Statistical MT tuning** has also been improved by using the pairwise approach of ranking with a classifier (Hopkins and May, 2011). In contrast to our approach, the works mentioned above do not use human judgments as material for the training process.

A couple of contributions (Ye et al, 2007; Duh, 2008) introduce the idea of using ranking in **MT evaluation** by developing a machine learning approach to train on human rank data, although these use reference translations and are only evaluated by producing an overall corpus-level ranking. State-of-the-art evaluation metrics such as METEOR (Lavie and Agarwal, 2007) also tune their parameters using human rankings. On the contrary, Avramidis et al (2011) do MT evaluation without references based on learned ranking, by using parsing features. Parton et al (2011) also show a version of their metric which achieves good correlation with human judgments just by analyzing target-language fluency using a language correction software.

The current work combines ideas from many previous works. We use human rankings for training and evaluating, we employ complex features and we measure effectiveness in two applications: sentence-selection and sentence-level evaluation.

3 Methods

3.1 Problem description

This work aims at developing a system for ranking multiple translation outputs. In detail, the system is given one source sentence and several translations which have been produced for this sentence, with the use of many MT systems. The goal is to derive several qualitative criteria over the translations and use them to order the translations based on their quality, i.e. to *rank* them.

In this ranking process, each translation is assigned an integer (further called a *rank*), which indicates its quality relatively to the competing translations for the same source sentence. E.g. given one source sentence and n translations for it, each of the latter would get a rank in the range $[1, n]$. The same rank may be assigned to two or more translation candidates, if the translations are of similar quality (i.e. there is no distinguishable difference between them). Such a case defines a *tie* between the two translation candidates.

This kind of qualitative ordering does not imply any absolute or generic measure of quality. Ranking takes place on a sentence level, which means that the inherent mechanism focuses on only one sentence at a time, considers the available translation options and makes a decision. Any assigned rank has therefore a meaning only for the sentence-in-focus and given the particular alternative translation candidates.

Finally, one further assumption as part of the current problem specification is that our system is not bound to the MT systems providing the outputs. This means that the usually small number of alternative translations may derive from many more MT engines with different characteristics and internal behaviour. The systems are therefore seen as *black boxes* and their translation outputs are treated on a merely superficial level, i.e. without any further information of how they were produced. Thus, we assume that the source and translated text contain enough information for assessing translation quality, approaching the way the task would be probably perceived by a human annotator.

3.2 Pairwise machine learning

The problem sketched above is treated as a typical machine learning problem. A ranker is *learned* from training material containing existing human rankings. The learning process results in a statistical model. This model can later reproduce the same task on unknown sentences or test data. Whereas the setup and the evaluation of the system takes place on a ranking level, for the core of the decision-making mechanism we follow the principle of decomposing full ranks in pairwise comparisons (Herbrich et al, 1999; Hüllermeier et al, 2008). Then, given one pair of translation candidates at a time, a classifier has to predict a binary decision on whether one translation candidate is better than the other.

In this context we train one classifier for the entire data set. Each ranking of n candidate translations is decomposed into $n \times (n-1)$ pairs of all possible combinations of two system outputs with replacement. Each of the resulting pairs is a training instance for the classifier and consists of a class value c and a set of features $(f_1, \ldots, f_n)$. For the pairwise comparison of two translation candidates t_i, t_j with human ranks r_i and r_j respectively, the class value is therefore set as:

$$c_{i,j} = \begin{cases} 1 & r_i < r_j \\ -1 & r_i > r_j \end{cases}$$

The approach of pairwise comparisons is chosen because it poses the machine learning question in a much simpler manner. Instead of treating a whole list of ranks, the classifier has to learn and provide a binary (positive or negative) answer to the simple question "*which of these two sentences is better?*". This also provides the flexibility of experimenting with many machine learning algorithms for the classification, including those which only operate on binary decisions.

As explained already, ties may exist in the training material. However, ties that appear on a pairwise level have been filtered out, since they do not provide any useful information about the simple comparison explained above.

3.3 From pairs back to ranking

During the application of the statistical model on test data, data processing follows the same idea: The test instances are broken down to pairs of sentences and given to the classifier for a binary decision. Consequently there is a need to recreate a ranking list out of the binary pairwise classification decisions.

3.3.1 Hard rank recomposition

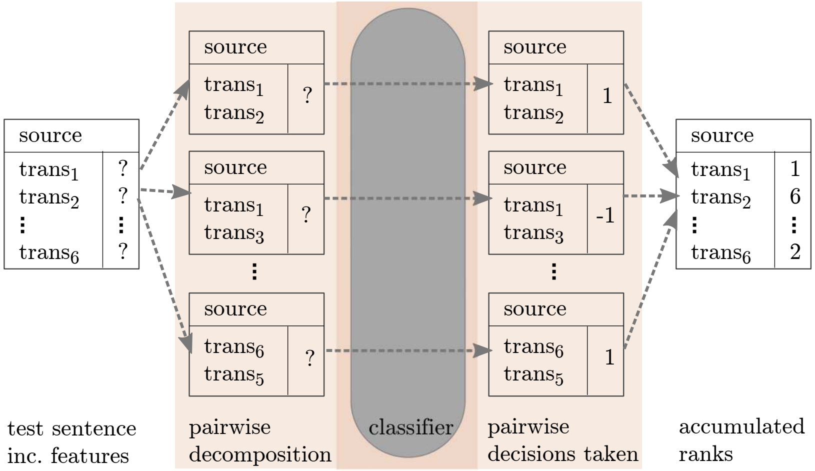

Fig. 1 The application of the statistical model, through the pairwise decomposition (left) and recomposition (right)

The simplest way to go ahead with this is to sum up the decisions of the classifier. For a number of n systems, following the previous notation, the rank r_i of translation t_i would be:

$$r_i = \sum_{j \neq i}^{n} c_{i,j}$$

The translation output which has "won" the most pairwise comparisons would get first on the list and then the outputs with fewer pairwise wins would follow accordingly (figure 1). We call this a *hard rank recomposition*, as only the binary decision of the classifier is taken into consideration upon summing up the predicted values.

3.3.2 Soft rank recomposition

One of the problems seen in previous work is that what we described here as a *hard rank recomposition* allows for the creation of ties. Indeed, the classifier may predict an equal number of wins for two or more translation outputs and therefore generate a tie among them. This may be intensified by the fact that the pairs have been generated in both directions, which would also result in a tie if the classifier is unable to distinguish the best out of two outputs but is forced to choose one of them.

However, the probabilistic set-up contains information which implies that not all classifier decisions are of "equal importance": statistical classifiers build their binary responses on a probabilistic basis. A translation output which has a number of wins with high certainty should be ranked higher than an output with an equal number of wins but with lower certainty. One can therefore use the probability of each decision to weigh the sum described in Section 3.3.1. This is thereof referred to as *soft recomposition.* This way, the rank r_i of translation t_i would be:

$$r_i = \sum_{j \neq i}^{n} p_{i,j} c_{i,j}$$

Since the probability $p_{i,j}$ is a decimal in the range of [0,1] as opposed to a binary value, it is expected that it reduces the cases where two translation outputs end up with an equal sum.

3.4 Feature acquisition

Similar to the previous works on quality estimation, the source sentence and the corresponding translations are analyzed by several linguistic tools, in order to provide a set of features indicative of the translation quality. Since one of the goals is not to be bound to the particular systems (Section 3.1) we consider only *black-box* features, i.e. derived solely from the text. The features used in this work fall into the following categories:

- **Parsing statistics**: One of the common issues that affect MT quality and acceptability is the grammaticality of the generated sentences. Such issues occur often in statistical systems (particularly the ones following the phrase-based approach) since they treat the generation process in a rather shallow way by using language models. As an additional measure of quality which can capture more complex phenomena (such as grammatical fluency, long distance structures, etc.) we include features derived from Probabilistic Context Free Grammars (PCFG) parsing (Petrov et al, 2006).

sentence	suggestion
Right after **hearing** about it, he described it as a “challenge”	*disambiguate -ing*
An fully comprehensive insurance with tax exemption	*'an'+consonant*
Tired and disappointed are **the the** fishermen	*repeating word*

Table 1 Sample corrections generated by rule-based language checking tools, observed in the development data

PCFG parsing operates by creating many possible tree parses for a given sentence, forming an n-best list of parse hypotheses. These hypotheses are scored probabilistically, leading to the selection of the tree with the highest overall probability. We allow an n-best list with a size of n=1000 and count **the number of trees generated**. Although for a majority of the sentences the n-best list reach the limit, some sentences have a smaller number of trees, which signifies fewer possible tree derivations, i.e. less parsing ambiguity, a feature which would be useful for our purpose.

Additionally, we extract and include the basic parsing statistics of the **parse log-likelihood**, the **confidence for the best parse tree** and the **average confidence of all trees**, as their values may reflect grammatical errors (Wagner and Foster, 2009).

- **Tree label counts**: In an effort to derive adequacy features, we rely on the assumption of isomorphism, i.e. the fact that the same or similar grammatical structures should occur on both source sentence and translation(s). Therefore, we count the basic node labels of the parse tree, namely the NPs, VPs, PPs, verbs, nouns, sentences, subordinate clauses and punctuation occurrences. The source and target equivalents of labels are manually matched so that their ratios could also be calculated. For example, the failure to properly translate a Verb Phrase should be indicated by an disproportional ratio.
- **Language checking**: Automatic rule-based *language quality checking*, similar to the one integrated on word processors, is applied on source and target sentences. This analysis (Siegel, 2011) provides a wide range of quality suggestions concerning **style**, **grammar** and **terminology** (see Table 1) and the corresponding quality scores. Since the individual occurrences of particular rules are rather sparse, we also sum the occurrences of the suggestions per category and in total.
- **Language-model probabilities**: Language models provide statistics on how likely the sequences of the words are for a particular language, so they are also an indication of fluency. From this category we mainly use the smoothed n-gram probability of the sentence.
- **Contrastive evaluation scores**: Each translation is scored with an automatic metric (e.g. Papineni et al, 2002), using the competitive translations as references. This has shown to perform well as a feature in similar tasks (Soricut et al, 2012).
- **Count-based features**: These are features which include the count of tokens, the average count of characters per token and the unknown words

Keeping the isomorphism assumption, an additional hint for the adequacy of the translation is applied for the features that appear in both source and target: The ratio of these features is calculated by dividing the feature value of each one of the translation outputs with the respective feature value of the source.

3.5 Machine learning algorithms

Since the core machine learning of the system operates with pairwise decisions, it is possible to use several machine learning algorithms:

- **Naïve Bayes** predicts the probability of a binary class c given a set of features

  $$p(c, f_1, \ldots, f_n) = p(c) \prod_{n}^{i=1} p(f_i|c)$$

  $p(c)$ is estimated on relative frequencies of the training pairwise examples. Since we are using continuous features f, their probabilities $p(f_i|c)$ are estimated with the locally weighted linear regression LOESS (Cleveland, 1979).
  Naïve Bayes works under the assumption that the features are statistically independent, which we cannot guarantee however. It has the advantage that it offers good scalability for the training process, given large data sets.
- The **k-nearest neighbour** (knn) algorithm classifies the test instances along with the closest training examples in the search space (Coomans and Massart, 1982). Unlike Naïve Bayes, there are no a priori assumptions about the distributions of the training data. However, a choice for the number (k) of the nearest neighbours is required, which is problem-specific. Here the common practice of setting the k equal to the square root of the number of training instances (Khedr, 2008) was adopted.
- **Logistic regression** operates with a maximization of a logistic function, producing values that range between zero and one (Cameron, 1998). In our case, the function is fitted using the Newton-Raphson algorithm to iteratively minimise the least squares error computed from training data (Miller, 2002), whereas the most useful features are selected with Stepwise Feature Selection (Hosmer, 1989). When compared to the previous algorithms, Logistic Regression typically demonstrates a better performance, as well as better handling of complex feature sets. On the other side, it has higher computational complexity and demands more time, which limits its applicability for exploring many experiment settings.

3.6 Evaluation

3.6.1 Classification performance

The first step of the evaluation considers the robustness of the learnt pairwise model and particularly its ability to reproduce the classification on many parts of the training set. We therefore compute the **Classification Accuracy** (CA), after performing cross-fold validation over the training set. This provides indications about the choice of the learning method and the feature set, but it is yet not suitable for evaluating the entire task of ranking.

3.6.2 Correlation with human judgments

The performance of automatic ranking is measured against human rankings. For this purpose we run a test set through the system and we measure the correlation of the produced rankings (one per sentence) with the original human rankings.

As a correlation metric we use **Kendall's tau** (Kendall, 1938; Knight, 1966), which measures the correlation between two ranking lists on a segment level, by counting *concordant* or *discordant* pairwise comparisons: For every sentence, the two rankings (machine-predicted and human) are decomposed into pairwise comparisons.[1] When the predicted comparison matches the respective one by the human annotator, we count a concordant pair, otherwise we count a discordant pair. Then, tau is computed by:

$$\tau = \frac{\text{concordant} - \text{discordant}}{\text{concordant} + \text{discordant}}$$

with values that range between minus one and one. This means that the ranking is better when the value gets closer to one.

The calculation follows the formula of the Workshop on Machine Translation (WMT; Callison-Burch et al, 2012), in order to be comparable with other methods: The test set is filtered, so that pairwise comparisons with reference translations are excluded from the calculations. Similarly, pairwise ties in the human-annotated test set are excluded from the calculations, as ties are considered to form uncertain samples that cannot be used for evaluation. For the remaining pairwise comparisons, where human annotation has not resulted in ties, every tie on the machine-predicted rankings is penalised by being counted as a discordant pair.

As the above calculation is defined on a segment (sentence) level, we thereof accumulate tau on the data set level in two ways:

[1] Decomposing again the previously recomposed ranking, instead of using the initially decomposed pairs (subsection 3.2), allows tau to compare the success of the recomposition methods (subsections 3.3.1 and 3.3.2)

- **Micro-averaged tau** (τ_μ) where concordant and discordant counts from all segments (i.e. sentences) are gathered and the fraction is calculated with their sums.[2]
- **Macro-averaged tau** (τ_m) where tau is calculated on a segment level and then averaged over the number of sentences. This shows equal importance to each sentence, irrelevant of the number of alternative translations.

3.6.3 Success of Sentence Selection

Whereas the tau coefficient describes the ability of a system to perform full ranking, one may be interested in the ability of the system to choose only the best sentence (e.g. for system combination).

For each sentence we look at the translation alternative ranked first by our system and we derive the rank that this translation has been assigned by humans. We repeat this for all the sentences of the test set and we count how many sentence selections were given each one of the corresponding human rank labels. Finally, we average this sum over the total number of sentences. An absolutely successful system would get a 100% ratio on the first human rank (i.e. all predicted first ranks being also chosen first by the annotators). Nevertheless, since the choices are highly subjective, it is expected that many selected sentences are also ranked lower by the humans; so the sentence selection is better, when more automatically selected sentences are given a higher rank by the humans.

4 Experiment

4.1 Development strategy

Due to the high amount of features and machine learning options, we face an exponential number of possible experiment parameters. However, in order to be able to draw some conclusions in a reasonable amount of time, we follow an incremental approach: first, we devise some feature sets that have shown to perform well in previous work (section 4.4.1) and also provide a focus on both source and translation. We use these feature sets for learning and testing with the default parameters of all the available methods. At a second phase, we repeat the experiments with modifications of the most successful parameter set, e.g. by slightly changing features or adding promising new ones.

Although this approach may not result into the optimal feature set, because of local maxima, it is sufficient to get a functioning model confirming the original idea. The best systems discovered through this development process are consequently used for further comparisons and conclusions.

[2] τ_μ is the tau calculation that appears in WMT results

4.2 Data

The experiment is set on human rankings resulting from the MT evaluation tasks by the Workshops on Machine Translation (Callison-Burch et al, 2008, 2009, 2010, 2011) for translating from German to English. For the development phase (Table 2), we use the evaluation data from years 2008, 2010[3] and 2011 as a training set, and 2009 as a test set.[4] At a later stage, we repeat the successful parameters of the development with various combinations of the data sets above, as training and test sets, in order to prove that the learnt models are widely applicable and do not overfit the development set.

In their original form, the data contain 1482 sentences. Each of these has been translated by up to 14 different MT systems. These 14 outputs were grouped randomly into batches of five translations, which are distributed also randomly to different annotators, providing a collection of 5366 batches of 5-rank judgments. Because of that, the comparison of two particular translation alternatives may have been evaluated many times by different humans, resulting in contradictory judgments.

We choose to ignore this upon training, hoping that the learning algorithms, due to their probabilistic nature, would not be affected by the contradictory overlaps.[5] However, concerning testing, a more robust point of reference is required: a learning method should not be penalised for making decisions on data points that humans anyway disagree. For these purpose, we merge the multiple batches of the same source sentence into one, so that each system output appears once in the new ranking. The system outputs are now ordered based on how many pairwise comparisons they won.[6] Contradictory pairwise judgments are eliminated through majority voting. Ties and cases of equal disagreement are removed.[7]

4.3 Implementation

N-gram features are computed with the SRILM toolkit (Stolcke, 2002) with an order of 5, based on monolingual training material from Europarl (Koehn, 2005) and News Corpus (version 6, Callison-Burch et al, 2011). PCFG parsing features are generated on the output of the Berkeley Parser (Petrov and Klein, 2007), trained over an English and a German treebank. The *Acrolinx IQ*[8] is used to annotate source and target sentences with language checking suggestions and provide style, grammar and spelling scores. The annotation

[3] In all of the experiments we exclude the crowdsourced sentences of 2010

[4] Classification accuracy on Table 2 is calculated with cross-validation on the training set

[5] Nevertheless, annotator disagreement is a factor that could increase the data noise

[6] Reducing the multiple ranking spans into one ranking has been lately an issue of discussion, as recent criticism advocates solving that as a *tournament* (Lopez, 2012). At the moment we still follow the standard way it was done by WMT until the year 2012.

[7] The processed data sets can be found at `http://www.dfki.de/~elav01/download/mtj12`

[8] `http://www.acrolinx.com` (proprietary)

model			hard recomposition				soft recomposition			
feat	classifier	CA%	τ_μ	τ_m	st%	tp%	τ_μ	τ_m	st%	tp%
#1	kNN	57.0	0.05	0.00	35.6	5.8	0.10	0.08	0.3	0.1
	Naive	57.8	-0.03	-0.07	34.0	13.7	0.12	0.14	1.9	0.3
#2	kNN	**60.7**	**0.10**	**0.12**	28.7	7.1	**0.18**	**0.22**	0.3	0.1
	Naive	**59.6**	**0.12**	**0.11**	13.8	3.4	**0.17**	**0.18**	1.9	0.2
#3	kNN	55.8	-0.06	-0.06	20.7	5.6	0.00	0.00	0.0	0.0
	LogReg	55.1	0.06	0.04	3.5	0.6	0.06	0.04	2.9	0.5
	Naive	54.9	0.02	0.05	3.7	0.7	0.02	0.05	2.7	0.6
#4	kNN	56.3	-0.04	-0.02	26.3	5.8	0.04	0.05	0.0	0.0
	LogReg	55.0	0.06	0.03	8.5	1.1	0.06	0.04	2.7	0.5
	Naive	55.2	0.00	0.04	4.8	0.8	0.01	0.04	2.4	0.5
#2.1	kNN	58.4	0.09	0.08	30.9	5.6	0.16	0.16	0.0	0.0
	LogReg	**61.5**	**0.24**	**0.27**	11.4	1.6	**0.25**	**0.29**	2.9	0.5
	Naive	59.8	0.16	0.15	20.2	4.3	0.21	0.21	2.9	0.4
#2.2	kNN	58.5	0.06	0.08	29.5	5.2	0.12	0.14	0.0	0.0
	LogReg	**61.5**	0.24	0.27	1.9	0.4	0.26	0.28	3.5	0.5
	Naive	59.6	0.15	0.13	24.7	5.1	0.20	0.19	3.2	0.5
#2.3	LogReg	61.4	0.24	0.26	9.3	1.3	0.25	0.27	6.1	0.7
	Naive	**61.0**	0.22	0.26	6.7	0.9	**0.23**	**0.28**	5.6	0.8
#2.4	kNN	59.4	0.06	0.08	27.1	5.5	0.12	0.16	0.0	0.0
	LogReg	61.3	**0.26**	**0.28**	3.5	0.7	**0.27**	**0.30**	2.9	0.4
	Naive	60.8	0.20	0.21	7.7	2.3	0.24	0.26	2.7	0.4

Table 2 On the vertical dimension, search of the most promising feature sets, done in two stages 4.4.1. On the horizontal dimension, improvement by soft recomposition. See subsections 3.6 and 4.4.2 for a description of the metrics

process is organised with the Ruffus library (Goodstadt, 2010) and the learning algorithms are executed using the Orange toolkit (Demšar et al, 2004).

4.4 Results

4.4.1 Best combination of feature set and learning algorithm

The search through different combinations of feature sets and classification methods is depicted in Table 2. Feature sets 1 - 4 derive from previous work (Soricut et al, 2012; Avramidis et al, 2011; Specia et al, 2012)[10] and are detailed in Table 3. Out of these, it appears that feature set 2 is the most successful one for this particular problem, providing a correlation which is acceptable to begin with. Knn slightly outperforms Naïve Bayes.

Consequently, experimentation considers extensions to feature set 2. Feature set 2.1 gives an improved combination when using logistic regression. It derives from the same annotation as feature set 2, with the difference that the features of the target had not been divided with the features of the source in order to provide a fixed ratio as a feature; instead, these features were given

[9] As *unknown words* we define the ones not seen in the monolingual corpus used for building the language model of the respective language

[10] We tried to come as close as possible to the original features sets when not all features were technically available

#1	source:	avg. characters per word, tri-gram probability, count of tokens, NPs
	target:	parse log-likelihood, count of unknown words[9], ratio of VPs, ratio of PPs, NPs, verbs, ratio of tokens count (Specia et al, 2012)
#2	source:	count of unknown words
	target:	count of unknown words, tokens ratio, ratio of parse trees, ratio of VPs, ratio of parse log-likelihood (Avramidis et al, 2011)
#3	source:	count of unknown words, tokens, dots, commas, avg. characters per word, LM probability
	target:	contrastive-BLEU, LM probability (SVR Model from Soricut et al (2012))
#4	source:	count of unknown words, tokens, dots, commas, avg. characters per word, LM probability
	target:	contrastive-BLEU, LM probability (M5P model from Soricut et al (2012))
#2.1	source:	count of unknown words, tokens, parse trees, VPs, parse log-likelihood
	target:	count of unknown words, tokens, parse trees, VPs, parse log-likelihood (same as #2 with no ratios)
#2.2	source:	count of unknown words, tokens, parse trees, VPs, NPs, parse log-likelihood
	target:	count of unknown words, tokens, parse trees, VPs, NPS, parse log-likelihood (same as #2.1 including NPs)
#2.3	source:	count of unknown words, tokens, parse trees, dots, commas, spelling score, grammar score, style score
	target:	contrastive-METEOR, count of unknown words, tokens, parse trees, dots, commas, spelling score, grammar score, style score
#2.4	source:	count of unknown words, tokens, parse trees, VPs, parse log-likelihood
	target:	contrastive-METEOR, count of unknown words, tokens, parse trees, VPs, parse log-likelihood (same as #2.1 with contrastive-METEOR)

Table 3 Description of the feature-sets used

separately. This is because logistic regression can learn a logistic function using two individual features, which may be more effective than their ratio.

Adding NP counts (feature set 2.2) does not show any improvement. Replacing parsing probability with spelling, grammar and style scores achieves some improvement, particularly for Naïve Bayes, which has its highest tau coefficient here.

The most successful feature set is 2.4, which extends 2.1: In a model learned with logistic regression (LogReg#2.4), it includes the number of unknown words, sentence length, the number of alternative parse trees, the count of VPs and the parse log-likelihood, but also a contrastive METEOR score for each one of the target sentences, using the others as pseudo-references. The best approaches are confirmed to generalise when applied on other data set combinations (Table 4).

In Table 5 we show the parse tree features and particularly how useful the parse log-likelihood along with the number of best-trees were. It seems that they both contribute to improving the tau correlation of the model.

	training-sets	test-set	CA	τ_μ
LogReg#2.1	2008,2010,2011	2009	62%	0.25
	2008,2009,2010	2011combo	64%	0.23
	2008,2009,2011	2010	59%	0.29
LogReg#2.4	2008,2010,2011	2009	62%	0.27
	2008,2009,2010	2011combo	65%	0.24
	2008,2009,2011	2010	58%	0.27

Table 4 The best methods perform equally well when applied on other training/test set-ups

	#2.4	#2.1
count of unknown trees, tokens, VPs,		
+parse loglikelihood	0.25	0.24
+number of parse trees	0.21	0.20
+number of parse trees, parse loglikelihood	0.27	0.25

Table 5 Correlations (τ_μ) achieved by using the numerical PCFG parsing features on the two most successful models (both using logistic regression with soft recomposition) for the development data set

test-set	2009	2010	2011combo
LogReg#2.1	0.26	0.29	0.23
LogReg#2.4	0.27	0.27	0.24
SmoothBLEU	-0,23	-0,16	-0,25
METEOR	0.20	0.30	0.12
Levenshtein	0.18	0.26	0.07

Table 6 Comparison of our best result with state-of-the-art reference-aware automatic metrics concerning correlation with human judgments (τ_μ). For the corresponding training sets refer to Table 4

4.4.2 Improvement by soft recomposition

The contribution of the soft recomposition of the rank (section 3.3.2) can be read off Table 2 on the horizontal dimension. The ties are measured by the percentage of the sentences which contain ties (st%) and the percentage of pairwise comparisons which are tied (tp%). The soft recomposition achieves higher tau correlation coefficients and significantly less ties for all the systems and particularly for the ones which show a positive correlation. In the best cases, using soft recomposition improves the correlation numbers by 40-80%. All further experiment results (Tables 4, 5, 6) are shown only for soft recomposition.

4.4.3 Comparison with state-of-the-art MT evaluation

Although our method uses no reference translations, it is still aimed at MT evaluation. Therefore, in lack of openly available competitors of its kind, it makes sense to compare its performance with automatic state-of-the-art MT metrics, to whom reference translations are available. Sentence-level smoothed BLEU, Levenshtein distance (Levenshtein, 1966) and METEOR (Lavie and Agarwal, 2007) were used. The results (Table 6) show that even without refer-

test-set	ranked with	rank1	rank2	rank3	rank4	rank5	rank6-14
2009	LogReg#2.1	34.3%	28.2%	15.2%	11.2%	6.7%	4.5%
	LogReg#2.4	34.6%	26.9%	15.2%	12.5%	7.5%	3.5%
	SmoothBLEU	27.1	26.1%	21.3%	15.7%	2.9%	7.2%
	METEOR	39.6%	31.4%	14.6%	9.6%	1.1%	3.7%
	Levenshtein	40.2%	27.7%	16.0%	10.4%	1.6%	4.3%
2010	LogReg#2.1	38.6%	26.0%	15.9%	8.2%	4.7%	6.5%
	LogReg#2.4	37.5%	26.2%	16.6%	8.2%	4.0%	7.5%
	SmoothBLEU	34.9%	23.7%	19.1%	11.9%	3.3%	7.5%
	METEOR	48.1%	27.2%	11.2%	8.1%	2.3%	2.8%
	Levenshtein	43.7%	25.6%	16.7%	7.2%	3.3%	3.3%
2011combo	LogReg#2.1	37.3%	22.5%	18.3%	11.3%	7.8%	2.8%
	LogReg#2.4	32.4%	26.1%	21.1%	12.7%	4.2%	3.5%
	SmoothBLEU	29.6%	20.4%	17.6%	17.6%	11.3%	3.3%
	METEOR	33.8%	26.8%	22.5%	10.6%	6.3%	0.1%
	Levenshtein	33.8%	29.6%	19.7%	12.0%	4.2%	0.8%

Table 7 Distribution of the human ranks for the selected sentence and comparison with reference-aware metrics

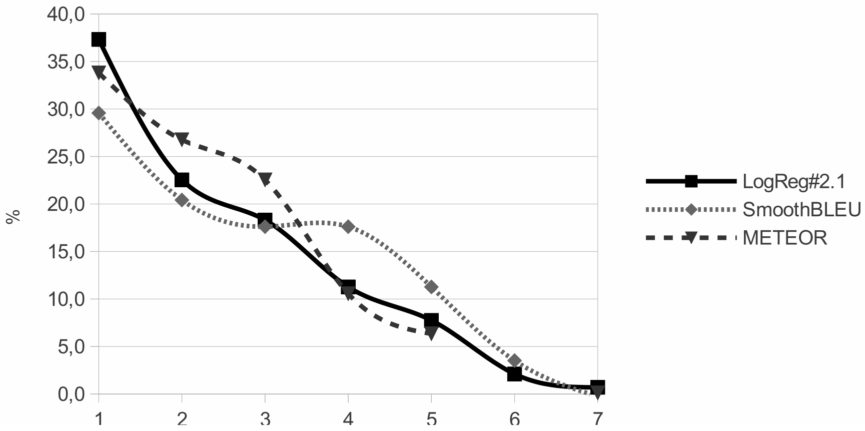

Fig. 2 Sample graphical representation of the mass distribution of the human ranks, corresponding to the selected sentences by the ranking mechanism. In this case (test 2011combo), our model fitted with logistic regression outperforms METEOR

ences, the correlation of our best systems with human judgment is comparable (test set 2010) or higher than the automatic metrics. Moreover, in the case of test set 2011combo, the correlation of LogReg#2.4 is double than that of METEOR.

4.4.4 Quality of Sentence Selection

We examine the quality of the automatically selected sentences (ranked first as explained in section 3.6.3). The difficulty of the problem is illustrated by the fact that even for the reference-aware metrics, only a 27-48% of the selected sentences have been ranked first by the human annotators (Table 7). However, these are relative indications, so in many cases rank 2 may be quite close in terms of quality. The performance of our methods is in principal comparable with that of the reference-aware metrics, while our method LogReg#2.1 performs again better than the automatic metrics on test set 2011combo. The better performance over 2011combo may be attributed to the fact that in contrast to the other test sets, it only contains results of statistical system combination and lacks direct output of rule-based systems, whose quality is harder to predict.

5 Conclusion

Machine learning was successfully used as part of a mechanism that is able to perform preference ranking on alternative machine translation outputs. Correlation with human judgments indicates promising results in building a mechanism which performs ranking, since its performance is comparable or higher than other state-of-the-art reference-aware automatic metrics.

The fact that ranking was decomposed into pairwise decisions allowed the integration of several machine learning algorithms with positive results. The recomposition of a ranking from pairwise decisions faced the problem of creating too many ties as a result of unclear and contradictory pairwise decisions. This was solved by weighing classification decisions with their prediction probabilities.

The best system uses logistic regression with a feature set that includes the number of unknown words, the sentence length, a contrastive METEOR score, and parsing statistics such as number of alternative parse trees, count of VPs and the parse log-likelihood.

Acknowledgments

This work has been developed within the TaraXŰ project, financed by TSB Technologiestiftung Berlin – Zukunftsfonds Berlin, co-financed by the European Union – European fund for regional development. Many thanks to Prof. Hans Uszkoreit for the supervision, Dr. Aljoscha Burchardt, Dr. Maja Popovič and Dr. David Vilar for their useful feedback, to Prof. Melanie Siegel for her support concerning the language checking tool and to Lukas Poustka for his technical help on feature acquisition.

References

Avramidis E (2011) DFKI System Combination with Sentence Ranking at ML4HMT-2011. In: Proceedings of the International Workshop on Using Linguistic Information for Hybrid Machine Translation and of the Shared Task on Applying Machine Learning Techniques to Optimising the Division of Labour in Hybrid Machine Translation, Barcelona, Spain, pp 99–103

Avramidis E, Popovic M, Vilar D, Burchardt A, Popović M (2011) Evaluate with Confidence Estimation : Machine ranking of translation outputs using grammatical features. In: Proceedings of the Sixth Workshop on Statistical Machine Translation, Edinburgh, UK, pp 65–70

Blatz J, Fitzgerald E, Foster G, Gandrabur S, Goutte C, Kulesza A, Sanchis A, Ueffing N (2004) Confidence estimation for machine translation. In: Proceedings of the 20th international conference on Computational Linguistics, Stroudsburg, PA, USA

Callison-Burch C, Fordyce C, Koehn P, Monz C, Schroeder J (2007) (Meta-) Evaluation of Machine Translation. In: Proceedings of the Second Workshop on Statistical Machine Translation, Prague, Czech Republic, pp 136–158

Callison-Burch C, Fordyce C, Koehn P, Monz C, Schroeder J (2008) Further Meta-Evaluation of Machine Translation. In: Proceedings of the Third Workshop on Statistical Machine Translation, Columbus, Ohio, pp 70–106

Callison-Burch C, Koehn P, Monz C, Schroeder J (2009) Findings of the 2009 Workshop on Statistical Machine Translation. In: Proceedings of the Fourth Workshop on Statistical Machine Translation, Athens, Greece, pp 1–28

Callison-Burch C, Koehn P, Monz C, Peterson K, Przybocki M, Zaidan O (2010) Findings of the 2010 Joint Workshop on Statistical Machine Translation and Metrics for Machine Translation. In: Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR, Uppsala, Sweden, pp 17–53

Callison-Burch C, Koehn P, Monz C, Zaidan O (2011) Findings of the 2011 Workshop on Statistical Machine Translation. In: Proceedings of the Sixth Workshop on Statistical Machine Translation, Edinburgh, UK, pp 22–64

Callison-Burch C, Koehn P, Monz C, Post M, Soricut R, Specia L (2012) Findings of the 2012 Workshop on Statistical Machine Translation. In: Proceedings of the Seventh Workshop on Statistical Machine Translation, Montréal, Canada, pp 10–51

Cameron A (1998) Regression analysis of count data. Cambridge University Press, Cambridge UK

Cleveland WS (1979) Robust locally weighted regression and smoothing scatterplots. Journal of the American statistical association 74(368):829–836

Coomans D, Massart D (1982) Alternative k-nearest neighbour rules in supervised pattern recognition. Analytica Chimica Acta (138):15–27

Demšar J, Zupan B, Leban G, Curk T (2004) Orange: From Experimental Machine Learning to Interactive Data Mining. In: Principles of Data Mining and Knowledge Discovery, pp 537–539

Duh K (2008) Ranking vs. Regression in Machine Translation Evaluation. In: Proceedings of the Third Workshop on Statistical Machine Translation, Columbus, Ohio, pp 191–194

Federmann C, Avramidis E, Ruiz MCj, van Genabith J, Melero M, Pecina P (2012) The ML4HMT Workshop on Optimising the Division of Labour in Hybrid Machine Translation. In: Proceedings of the 8th ELRA Conference on Language Resources and Evaluation, Istanbul, Turkey

Goodstadt L (2010) Ruffus: a lightweight Python library for computational pipelines. Bioinformatics 26(21):2778–2779

He Y, Ma Y, van Genabith J, Way A (2010) Bridging SMT and TM with Translation Recommendation. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, pp 622–630

Herbrich R, Graepel T, Obermayer K (1999) Support Vector Learning for Ordinal Regression. In: International Conference on Artificial Neural Networks, pp 97 – 102

Hopkins M, May J (2011) Tuning as ranking. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Edinburgh, UK, pp 1352–1362

Hosmer D (1989) Applied logistic regression, 8th edn. Wiley, New York [u.a.]

Hüllermeier E, Fürnkranz J, Cheng W, Brinker K (2008) Label ranking by learning pairwise preferences. Artificial Intelligence 172(16-17):1897–1916

Kendall MG (1938) A New Measure of Rank Correlation. Biometrika 30(1-2):81–93

Khedr AM (2008) Learning k-Nearest Neighbors Classifier from Distributed Data. Computing and Informatics 27(3):355–376

Knight WR (1966) A computer method for calculating Kendalls tau with ungrouped data. Journal of the American Statistical Association 61(314):436–439

Koehn P (2005) Europarl: A Parallel Corpus for Statistical Machine Translation. In: Conference Proceedings: the tenth Machine Translation Summit, AAMT, AAMT, Phuket, Thailand, pp 79–86

Lavie A, Agarwal A (2007) METEOR: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments. In: Proceedings of the Second Workshop on Statistical Machine Translation, Prague, Czech Republic, pp 228–231

Levenshtein V (1966) Binary Codes Capable of Correcting Deletions and Insertions and Reversals. Soviet Physics Doklady 10(8):707–710

Lopez A (2012) Putting Human Assessments of Machine Translation Systems in Order. In: Proceedings of the Seventh Workshop on Statistical Machine Translation, Montréal, Canada, pp 1–9

Miller A (2002) Subset Selection in Regression, 2nd edn. Chapman & Hall, London

Papineni K, Roukos S, Ward T, Zhu WJ (2002) BLEU: a Method for Automatic Evaluation of Machine Translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, pp 311–318

Parton K, Tetreault J, Madnani N, Chodorow M (2011) E-rating Machine Translation. In: Proceedings of the Sixth Workshop on Statistical Machine Translation, Edinburgh, UK, pp 108–115

Petrov S, Klein D (2007) Improved inference for unlexicalized parsing. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics, Rochester, New York, pp 404–411

Petrov S, Barrett L, Thibaux R, Klein D (2006) Learning Accurate, Compact, and Interpretable Tree Annotation. In: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Sydney, Australia, pp 433–440

Raybaud S, Lavecchia C, David L, Kamel S (2009a) Word-and sentence-level confidence measures for machine translation. In: 13th Annual Meeting of the European Association for Machine Translation, European Association of Machine Translation, Barcelona, Spain

Raybaud S, Lavecchia C, Langlois D, Kamel S (2009b) New Confidence Measures for Statistical Machine Translation. Proceedings of the International Conference on Agents pp 394–401

Rosti AV, Ayan NF, Xiang B, Matsoukas S, Schwartz R, Dorr BJ (2007) Combining Outputs from Multiple Machine Translation Systems. In: Proceedings of the North American Chapter of the Association for Computational Linguistics Human Language Technologies, Rochester, New York, pp 228–235

Sánchez-Martínez F (2011) Choosing the best machine translation system to translate a sentence by using only source-language information. In: Proceedings of the 15th Annual Conference of the European Association for Machine Translation, Leuve, Belgium, pp 97–104

Siegel M (2011) Autorenunterstützung für die Maschinelle Übersetzung. In: Multilingual Resources and Multilingual Applications: Proceedings of the Conference of the German Society for Computational Linguistics and Language Technology (GSCL), Hamburg

Soricut R, Narsale S (2012) Combining Quality Prediction and System Selection for Improved Automatic Translation Output. In: Proceedings of the Seventh Workshop on Statistical Machine Translation, Montréal, Canada, pp 163–170

Soricut R, Wang Z, Bach N (2012) The SDL Language Weaver Systems in the WMT12 Quality Estimation Shared Task. In: Proceedings of the Seventh Workshop on Statistical Machine Translation, Montréal, Canada, pp 145–151

Specia L, Turchi M, Cancedda N, Dymetman M, Cristianini N (2009) Estimating the Sentence-Level Quality of Machine Translation Systems. In: 13th Annual Meeting of the European Association for Machine Translation, Barcelona, Spain., pp 28–35

Specia L, Raj D, Turchi M (2010) Machine translation evaluation versus quality estimation. Machine Translation 24(1):39–50

Specia L, Felice M (2012) Linguistic Features for Quality Estimation. In: Proceedings of the Seventh Workshop on Statistical Machine Translation, Montréal, Canada, pp 96–103

Stolcke A (2002) SRILM – An Extensible Language Modeling Toolkit. In: Proceedings of the Seventh International Conference on Spoken Language Processing, pp 901–904

Ueffing N, Ney H (2005) Word-level confidence estimation for machine translation using phrase-based translation models. Computational Linguistics pp 763–770

Vilar D, Avramidis E, Popović M, Hunsicker S (2011) DFKI's SC and MT Submissions to IWSLT 2011. In: Proceedings of the International Workshop on Spoken Language Translation 2011, San Francisco, CA, USA, pp 98–105

Wagner J, Foster J (2009) The effect of correcting grammatical errors on parse probabilities. In: Proceedings of the 11th International Conference on Parsing Technologies, Stroudsburg, PA, USA, pp 176–179

Ye Y, Zhou M, Lin CY (2007) Sentence level machine translation evaluation as a ranking problem: one step aside from BLEU. In: Proceedings of the Second Workshop on Statistical Machine Translation, Association for Computational Linguistics, Prague, Czech Republic, pp 240–247