Comparative quality estimation: Automatic sentence-level ranking of multiple Machine Translation outputs

Eleftherios Avramidis
German Research Center for Artificial Intelligence (DFKI)
Language Technology Lab
Alt Moabit 91c, 10559 Berlin, Germany
eleftherios.avramidis@dfki.de

ABSTRACT
A machine learning mechanism is learned from human annotations in order to perform preference ranking. The mechanism operates on a sentence level and ranks the alternative machine translations of each source sentence. Rankings are decomposed into pairwise comparisons so that binary classifiers can be trained using black-box features of automatic linguistic analysis. In order to re-compose the pairwise decisions of the classifier, this work introduces weighing the decisions with their classification probabilities, which eliminates ranking ties and increases the coefficient of the correlation with the human rankings up to 80%. The authors also demonstrate several configurations of successful automatic ranking models; the best configuration achieves acceptable correlation with human judgments (tau=0.30), which is higher than that of state-of-the-art reference-aware automatic MT evaluation metrics such as METEOR and Levenshtein distance.

KEYWORDS: Machine Translation, Quality Estimation, Ranking.

Proceedings of COLING 2012: Technical Papers, pages 115–132,
COLING 2012, Mumbai, December 2012.

1 Introduction

As machine translation (MT) comes closer to the everyday use, the need to predict the quality of machine-translated text is being of intense interest. For this reason, research has been focusing more and more on developing methods of assessing the translated content and deriving indications of translation performance in a real-time translation environment, without access to the correct (reference) translations.

Here we are focusing on a specific scenario: we need an automatic system able to rank several machine translation outputs for one given source sentence, according to their comparative quality. This kind of ranking, performed by human annotators, has been established as a practice for evaluating MT-output. Therefore, we attempt to perform "machine ranking", by employing Machine Learning approaches in order to imitate the human behaviour. This is done through a statistical classifier, which is trained given existing human ranks and several qualitative criteria on the text.

This idea can serve several existing applications in MT, as it touches the fields of Hybrid MT, System Combination and MT Evaluation. The applicability is even broader, as the approach presented is system-independent and relies on generic automatic analysis applied on any input containing sets of one source and several translation outputs.

2 Previous work

Quality Estimation is a rather recent aspect in research on Machine Translation. As a field, it tries to provide quality assessment on the translation output without the availability of reference translations. Previous work includes statistical methods on predicting word-level confidence (Ueffing and Ney, 2005; Raybaud et al., 2009b), correctness of a sentence (Blatz et al., 2004) and has been recently evolved into a regression problem (Specia et al., 2009; Raybaud et al., 2009a) for estimating correctness scores or correctness probabilities.

Whereas the aforementioned work has been focusing on estimating absolute measures of quality for a single output, our focus is on the comparative estimation of quality among several system outputs. In this direction, Rosti et al. (2007) perform sentence-level selection with generalized linear models, based on re-ranking N-best lists merged from many MT systems. Sánchez-Martínez (2011) uses only source-language information in order to build a classifier which chooses which machine translation system should be used in order to translate a sentence. As an application to statistical MT tuning, Hopkins and May (2011) improve the tuning MERT process by using the pairwise approach of ranking with a classifier. Others (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.

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 rank data, though these are using reference translations and are only evaluated by producing an overall corpus-level ranking. METEOR, one of the state-of-the-art evaluation metrics (Lavie and Agarwal, 2007) also gets its components tuned over human rankings. Avramidis et al. (2011) do MT evaluation without references based on learned ranking, by using parsing features and Parton et al. (2011) also show a configuration of their metric which achieves good correlation with human judgments without any reference information, using target features produced by a language correction software.

Reported work has used various aspects of weighing or training over human ranking. Following

on a similar path, we are focusing on the ability of a mechanism to reproduce human preference rankings and compare its outcome with existing evaluation methods.

3 Methods

3.1 Problem description

As already introduced in Section 1, this work is aiming to a mechanism 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.

It should be thereof clear that such a 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 a bag of 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, one assumption is that the source and translated text contain enough information for assessing translation quality, probably approaching the way the task would be perceived by a human annotator.

3.2 Machine learning on pairwise decomposition of ranking

The problem is hereby treated as a typical machine learning paradigm. Ranking is *learned* from a 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 going pairwise (Herbrich et al., 1999; Hüllermeier et al., 2008): ranking lists are decomposed to pairwise comparisons. Then, given one pair of translation candidates at a time, a classifier has to predict a binary decision on whether the first translation candidate is better than the latter.

In this context, we train one classifier for the entire data set. Each ranking of n candidate translations is decomposed to $n \times (n-1)$ pairs of all possible combinations of two system outputs with replacement. Each of the resulting pairs forms a training instance for the classifier. Each training instance provided to the classifier consists of a class value c and a set of features $(f_1, \ldots, f_n)$ regarding the translation quality/preferences. 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 forms the machine learning question in a much simplified 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 gives the flexibility of experimenting with many machine learning algorithms for the classification, including those which only operate on a binary class.

As explained in Section 3.1, ties may exist in the training material. Though, ties that appear on a pairwise level have been filtered out, since this does not add any useful information on the simple comparison explained above. This means that the pairwise comparisons of the tied outputs with the other outputs are not filtered out; only those between the two tied systems are. A further handling of ties by introducing a third class or a cascade of two classifiers would be a possibility to investigate, but we leave this aside for the moment in order to test the basic functionality given the most promising properties. As we will see later in this section, the ties are treated rather as an uncertainty of the system for either of the classes.

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.

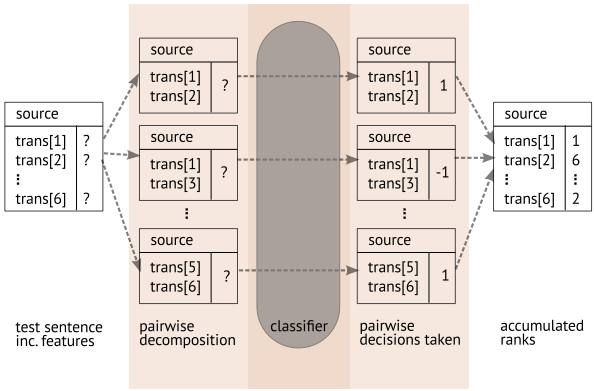

Figure 1: The application of the statistical model, through the pairwise decomposition (left) and recomposition (right)

3.3.1 Hard rank recomposition

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 annotation, 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 less 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, it is intuitive that the classifier may predict an equal number of wins for two or more translation outputs and therefore generate a tie among them. This may also be intensified by the fact that the pairs have been generated in both directions, which would also result into 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 more uncertain. One can therefore use the probability of each decision to weigh the sum described in Section 3.3.1. 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 more uncertain. 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 long 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 tools of linguistic analysis, in order to provide a set of features indicative of the translation quality. Since one of the goals is to not be bound to the systems participating (Section 3.1), these are *black-box* features, i.e. deriving solely from the text. The features used in this work fall into the following categories:

- **Count-based features**: count of tokens, average count of characters per token, count of unknown words
- **Parsing statistics**: One of the common issues that affect MT quality and acceptability is the grammaticality of the generated sentences. This is intense in many statistical systems

(particularly the ones following the phrase-based approach) since they treat the generation process in a rather shallow way. Most often language models are used within the MT systems in order to optimize the output for the highest probability of the consequent *n*-grams. 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).

PCFG parsing operates by creating many possible tree parses for a given a 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 allowed an n-best list with a size of n=1000 and counted **the number of trees generated**. Although the *n*-best list reaches the limit for the majority of the sentences, some sentences have a smaller number of trees, which signifies less possible tree derivations, i.e. less parsing ambiguity, a feature which would be useful for our use.

Additionally, we extracted and included the basic parsing statistics of the **overall parse log-likelihood**, the **confidence for the best parse tree** and the **average confidence of all trees**.

- **Tree label counts**: In an effort to derive some adequacy features, we relied on the assumption of isomorphism; i.e. the fact that the same or similar grammatical structures should occur on both source and translation(s). Therefore, we counted the basic node labels of the parse tree, namely the NPs, VPs, PPs, verbs, nouns, sentences, subordinate clauses and punctuation occurrencies. The source and target equivalents of labels were manually matched so that their ratios could also be calculated. E.g. the failure to properly translate a Verb Phrase should be indicated by an improportional ratio.

- **Language checking**: Source and target sentences were subject to automatic rule-based *language quality checking* (Siegel, 2011), providing a wide range of quality suggestions concerning **style**, **grammar** and **terminology**, summed up in relevant quality scores.

- **Language-model probabilities**: Language models are also an indication of fluency, since they provide statistics on how likely the sequences of the words are for a particular language. Although Statistical MT systems are expected to already optimize over the language model, as mentioned above, other types of systems may still benefit from this features. This feature category includes 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).

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

3.5 Machine learning algorithms

The modular approach of the pairwise classification allowed the use of several machine learning algorithms as part of the system core.

- **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 by relative frequencies of the training pairwise examples, while the probabilities $p(f_i|c)$ for the continuous features f are estimated with the locally weighted linear regression LOESS (Cleveland, 1979).

  Naïve Bayes has the drawback that it requires the features to be statistically independent. However, as an algorithm it can be trained pretty fast and scales well with large amount of data and big feature sets.

- The **k-nearest neighbour** (K-nn) algorithm performs classification to the closed training examples in the search space (Coomans and Massart, 1982). This algorithm does not have a priori assumptions about the distributions of the training data. Though, the method requires a choice for the number (k) of the nearest neighbors, which is problem-specific. For these experiments, we followed the common practice of setting the k equal to the square root of the number of training instances.

- **Logistic regression** tries to maximize a logistic function, whose values range between zero and one (Cameron, 1998). It was fitted using Newton-Raphson algorithm to iteratively minimize least squares error computed from training data (Miller, 2002), whereas Stepwise Feature Selection (Hosmer, 1989) was included. Logistic Regression generally performs better than the previous algorithms, though it has higher computational complexity and therefore its calculation is time-demanding, limiting the possibility to explore many experiment parameters.

3.6 Evaluation

3.6.1 Classification performance

As a classifier is the core part of the system, its robustness and ability to successfully take its binary decisions are indicative of the performance of the entire system. As a basic indications on the success of the learning process we compute **Classification Accuracy** (CA), as a result of cross-fold validation over the training set. This part is useful for evaluating the choice of the learning method and the feature set.

3.6.2 Correlation with human judgments

The system is tested on how well it can rank translation quality, compared to the ranking a human would do for this purpose. After building a model, the system is used in order to perform ranking on a test-set. This test-set has been excluded from the training data and provides human rankings which are hidden during the test classification, but are afterwards used for evaluating the success of the automatic process. The final goal is to measure the sentence-level correlation of the automatically produced rankings with the ones chosen by humans.

For the correlation measurement, we follow **Kendall's tau** coefficient (Kendall, 1938; Knight, 1966), which is suitable for measuring correlation between two ranking lists on a segment level: For every sentence, the machine-predicted and the human ranking are decomposed into

pairwise comparisons. Each automatic pairwise comparison is compared with the respective human comparison and it is counted as *concordant* or *discordant*, depending on whether these two comparisons agree or not. Tau is then given by the fraction:

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

with values that range between minus one and one, whereas the closer the value gets to one, the better the ranking is. The calculation follows the formula of the Workshop in Machine Translation (Callison-Burch et al., 2012), in order to be comparable with other methods: Pairwise comparisons with reference translations and pairwise ties in the human-annotated test-set are ignored. On the contrary, every tie on the machine-predicted rankings is penalized by being counted as a discordant pair.

We thereof present two versions of the tau:

- **Overall tau** (τ) where concordant and discordant counts from all segments (i.e. sentences) are gathered and the fraction is calculated with their sums
- **Average segment tau** ($\overline{\tau}_{\text{seg}}$) 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.

4 Experiment

4.1 Data sets

Both our training and test data were extracted from human-annotated data containing comparisons of the outputs of several German-to-English MT systems, as a result of the evaluation tasks run by the Workshops on Machine Translation (Callison-Burch et al., 2008, 2009, 2010, 2011), which have been freely available for further research. In the development phase, our training set consisted of the human rankings of years 2008, 2010[1], 2011 and the test-set from the year 2009. In order to re-assure that the system is not overfitting the development environment, the best systems were also tested upon a different set-up, where the human rankings of years 2008, 2009 and 2010 were used for training and the rankings from 2011 system combination task (2011c) were used for testing.

The provided data-sets contain human judgments organized in rankings of at most five sentence at a time. Therefore, the alternative translation outputs for one source sentence are often spanned along many 5-way rankings. For the test set, the multiple rankings of the same source sentence (produced by all available systems) were aggregated into one ranking, and the ties on both pairwise and ranking level were removed. One should also notice that repetitive human rankings of the same systems often disagree with each other, which signifies the existence of some noise in our data.

4.2 Implementation

PCFG parsing features were generated on the output of the Berkeley Parser, with the default grammars based on an English and a German treebank (Petrov and Klein, 2007). N-gram features were based on language models of order 5, built with the SRILM toolkit (Stolcke, 2002) on monolingual training material from the Europarl (Koehn, 2005) and the News (Callison-Burch et al., 2011) corpora. The *Acrolinx IQ*[2] was used to annotate source and target with language checking suggestions and provide style, grammar and spelling scores. The annotation process was organized with the Ruffus library (Goodstadt, 2010) and the learning algorithms were executed using the Orange toolkit (Demšar et al., 2004).

4.3 Strategy

The amount of features and learning options provide an exponential number of experiment parameters. However, in order to be able to draw a fair amount of conclusions in a decent amount of time, we followed an incremental approach: first, we devised some feature sets that have shown to perform well in previous work[3] and used them for learning and testing with the default parameters of all the available methods. Secondly, we repeated the experiments with variations of the most successful parameter set, e.g. by slightly modifying the features or adding promising new features. This approach may have stalled to local maxima, but it should suffice if it can provide a functioning system confirming the original idea.

[1] In all of the experiments we excluded the crowdsourced sentences contained in the set of 2010

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

[3] We tried to come as close to the original feature set when not all features were technically available

4.4 Results

4.4.1 Searching for the best system

The search through different combinations of feature sets and classification methods is depicted in Table 1. Feature sets 2 - 5 derive from previous work (Soricut et al., 2012; Avramidis et al., 2011; Specia et al., 2012) and are explained 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. *K-nn* slightly outperforms Naïve Bayes.

Consequently, extensions to feature set 2 are considered for further experimentation. Feature set 2.1 provides an improved combination with logistic regression: It derives from the same annotation as feature set 2, with the difference that the features of the target had not been not divided with the features of the source, in order to provide a fixed ratio as a feature; instead, these features were given separately. Due to its power to do a logistic search, we could assume that this learner treats better the factors of the ratio if given separately.

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

The most successful feature set is 2.4, which extends 2.1: Learned with logistic regression, 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 additionally a contrastive METEOR score.

4.4.2 Improvement by soft recomposition

A basic contribution of this paper is the introduction of the soft recomposition of the ranks. This is obvious by reading Table 1 on the horizontal dimension: the soft recomposition achieves higher taus 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%.

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

Although our method uses no reference translations, it still maintains the notion of 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 get available. Sentence-level smoothed-BLEU (Papineni et al., 2002), METEOR (Lavie and Agarwal, 2007) and Levenshtein distance (Levenshtein, 1966) were used. This comparison was also done on a set-up different to that of the development phase. The results (Table 2) show that even without references, the correlation of our best system with human judgment is at least 35% higher than that of the standard metrics.

model		CA	hard recomposition			soft recomposition		
feat	classifier		τ	$\overline{\tau}_{seg}$	ties	τ	$\overline{\tau}_{seg}$	ties
#1	kNN	57,0%	0,05	0,00	259	0,10	0,08	6
	Naive	57,8%	-0,03	-0,07	615	0,12	0,14	12
#2	kNN	**60,7%**	**0,10**	**0,12**	317	**0,18**	**0,22**	6
	Naive	**59,6%**	**0,12**	**0,11**	152	**0,17**	**0,18**	8
#3	kNN	55,8%	-0,06	-0,06	250	0,00	0,00	0
	LogReg	55,1%	0,06	0,04	29	0,06	0,04	23
	Naive	54,9%	0,02	0,05	30	0,02	0,05	26
#4	kNN	56,3%	-0,04	-0,02	261	0,04	0,05	0
	LogReg	55,0%	0,06	0,03	51	0,06	0,04	22
	Naive	55,2%	0,00	0,04	34	0,01	0,04	22
#2.1	kNN	58,4%	0,09	0,08	252	0,16	0,16	0
	LogReg	**61,5%**	**0,24**	**0,27**	72	**0,25**	**0,29**	24
	Naive	59,8%	0,16	0,15	194	0,21	0,21	20
#2.2	kNN	58,5%	0,06	0,08	233	0,12	0,14	0
	LogReg	**61,5%**	0,24	0,27	74	0,26	0,28	18
	Naive	59,6%	0,15	0,13	228	0,20	0,19	24
#2.3	Naive	**61,0%**	**0,22**	**0,26**	83	**0,23**	**0,28**	24
	LogReg	61,4%	0,24	0,26	60	0,25	0,27	31
#2.4	kNN	59,4%	0,06	0,08	249	0,12	0,16	0
	LogReg	61,3%	**0,26**	**0,28**	68	**0,27**	**0,30**	15
	Naive	60,8%	0,20	0,21	141	0,24	0,26	11

Table 1: Search of the most promising feature sets tested on the development test-set . Feature set #2 from previous work extended with additional features: Logistic Regression with feature sets #2.4 and #2.1 had the highest τ correlation with human rankings. Improvement by using *soft rank re-composition* (Section 3.3.2) is also illustrated

test-set	2009	2011c
SmoothBLEU	-0,23	-0,25
METEOR	0,20	0,12
Levenshtein	0,18	0,07
Machine Ranking (LogReg #2.4)	**0,27**	**0,24**

Table 2: Comparison of our best result with state-of-the-art reference-aware automatic metrics concerning correlation with human judgments (τ). The model is also succesfully tested in order to rank wmt2011-combo translation options (excluded from training)

#1	source:	avg. characters per word, tri-gram probability, count of tokens, NPs
	target:	parse log-likelihood, count of unknown words, 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

5 Conclusion

Machine learning was successfully used as part of a mechanism which is able to perform preference ranking on alternative machine translation outputs. Correlation with human judgments indicates a success in building a mechanism which performs ranking, since even without access to reference information, its performance is higher than other state-of-the-art reference-aware 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 was facing 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, sentence length, a contrastive METEOR score, and parse 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 Aljoscha Burchard, Hans Uszkoreit and David Vilar for their useful feedback, to Lukas Poustka for his technical help on feature acquisition and to Melanie Siegel for her support concerning the language checking tool.

References

Avramidis, E., Popovic, M., Vilar, D., Burchardt, A., and 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*, pages 65–70, Edinburgh, Scotland. Association for Computational Linguistics.

Blatz, J., Fitzgerald, E., Foster, G., Gandrabur, S., Goutte, C., Kulesza, A., Sanchis, A., and Ueffing, N. (2004). Confidence estimation for machine translation. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Callison-Burch, C., Fordyce, C., Koehn, P., Monz, C., and Schroeder, J. (2008). Further Meta-Evaluation of Machine Translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 70–106, Columbus, Ohio. Association for Computational Linguistics.

Callison-Burch, C., Koehn, P., Monz, C., Peterson, K., Przybocki, M., and 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*, pages 17–53, Uppsala, Sweden. Association for Computational Linguistics.

Callison-Burch, C., Koehn, P., Monz, C., Post, M., Soricut, R., and Specia, L. (2012). Findings of the 2012 Workshop on Statistical Machine Translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 10–51, Montréal, Canada. Association for Computational Linguistics.

Callison-Burch, C., Koehn, P., Monz, C., and Schroeder, J. (2009). Findings of the 2009 Workshop on Statistical Machine Translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 1–28, Athens, Greece. Association for Computational Linguistics.

Callison-Burch, C., Koehn, P., Monz, C., and Zaidan, O. (2011). Findings of the 2011 Workshop on Statistical Machine Translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 22–64, Edinburgh, Scotland. Association for Computational Linguistics.

Cameron, A. (1998). *Regression analysis of count data*. Cambridge University Press, Cambridge UK; New York NY USA.

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

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

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

Duh, K. (2008). Ranking vs. Regression in Machine Translation Evaluation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 191–194, Columbus, Ohio. Association for Computational Linguistics.

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

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

Hopkins, M. and May, J. (2011). Tuning as ranking. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '11)*, pages 1352–1362, Edinburgh, Scotland. Association for Computational Linguistics Morristown, NJ, USA.

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

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

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

Knight, W. R. (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. *MT Summit*, 5.

Lavie, A. and 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*, pages 228–231, Prague, Czech Republic. Association for Computational Linguistics.

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

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

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

Parton, K., Tetreault, J., Madnani, N., and Chodorow, M. (2011). E-rating Machine Translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 108–115, Edinburgh, Scotland. Association for Computational Linguistics.

Petrov, S., Barrett, L., Thibaux, R., and 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*, pages 433–440, Sydney, Australia. Association for Computational Linguistics.

Petrov, S. and Klein, D. (2007). Improved inference for unlexicalized parsing. In *Proceedings of Human Language Technologies: The 2007 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-2007)*. Association for Computational Linguistics.

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

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

Rosti, A.-V., Ayan, N. F., Xiang, B., Matsoukas, S., Schwartz, R., and Dorr, B. J. (2007). Combining Outputs from Multiple Machine Translation Systems. In *Proceedings of the North American Chapter of the Association for Computational Linguistics Human Language Technologies*, pages 228–235, Rochester, New York. Association for Computational Linguistics.

Sánchez-Martínez, F. (2011). Choosing the best machine translation system to translate a sentence by using only source-language information. In Forcada, M. L., Depraetere, H., and Vandeghinste, V., editors, *Proceedings of the 15th Annual Conference of the European Associtation for Machine Translation*, number May, pages 97–104, Leuve, Belgium. European Association for Machine Translation.

Siegel, M. (2011). Autorenunterstützung für die Maschinelle Übersetzung. In Hedeland, H., Schmidt, T., and Wörner, K., editors, *Multilingual Resources and Multilingual Applications: Proceedings of the Conference of the German Society for Computational Linguistics and Language Technology (GSCL)*, Hamburg.

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

Soricut, R., Wang, Z., and 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*, pages 145–151, Montréal, Canada. Association for Computational Linguistics.

Specia, L., Street, S., Court, R., and Felice, M. (2012). Linguistic Features for Quality Estimation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 96–103, Montréal, Canada. Association for Computational Linguistics.

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

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

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

Vilar, D., Avramidis, E., Popović, M., and 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.

Ye, Y., Zhou, M., and Lin, C.-Y. (2007). Sentence Level Machine Translation Evaluation as a Ranking. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 240–247, Prague, Czech Republic. Association for Computational Linguistics.