PBML

The Prague Bulletin of Mathematical Linguistics

NUMBER 100 OCTOBER 2013 63–72

RankEval: Open Tool for Evaluation of Machine-Learned Ranking

Eleftherios Avramidis

Language Technology Lab
German Research Center for Artificial Intelligence (DFKI)

Abstract

Recent research and applications for evaluation and quality estimation of Machine Translation require statistical measures for comparing machine-predicted ranking against gold sets annotated by humans. Additional to the existing practice of measuring segment-level correlation with Kendall tau, we propose using ranking metrics from the research field of Information Retrieval such as Mean Reciprocal Rank, Normalized Discounted Cumulative Gain and Expected Reciprocal Rank. These reward systems that predict correctly the highest ranked items than the one of lower ones. We present an open source tool ”RankEval” providing implementation of these metrics. It can be either run independently as a script supporting common formats or can be imported to any Python application.

1. Introduction

Research in Machine Translation (MT) has resulted into the development of various Machine Translation systems over the years. One of the most prominent ways of assessing their performance is to do it comparatively, i.e. comparing them and ordering them in terms of quality. This offers the possibility to be consistent with human quality judgements, without having to rely on underspecified "absolute" quality numbers, which are often hard to define and derive objectively.

The result of ordering translations in terms of their quality has had a few applications focusing on a sentence level. One of these applications refers to assessing the quality of automatic evaluation metrics. In particular, since quite a few years, the Evaluation Shared Task of the Workshop on Machine Translation (Callison-Burch et al., 2008) has used the so-called "segment-level" ranking, in order to compare rank-

Corresponding author: eleftherios.avramidis@dfki.de
Cite as: Eleftherios Avramidis. RankEval: Open Tool for Evaluation of Machine-Learned Ranking. The Prague Bulletin of Mathematical Linguistics No. 100, 2013, pp. 63–72. doi: 10.2478/pralin-2013-0012.

ings produced by automatic evaluation metrics against the ones devised by human annotators. In most cases, translation segments are defined by periods, roughly as long as one sentence.

Additionally, the use of several Machine Translation (MT) systems within translation workflows pretty often requires automatic Quality Estimation systems that predict the ranking of the translation quality on a sentence level. The performance of such Quality Estimation rankers can be assessed when sentence-level ranking lists are compared with the ranking a human would do.

In both above tasks, predicted ranking is evaluated against the human ranking, using calculations following the Kendall tau correlation coefficient. On top of that, in this paper we present some existing measures that have been used in other fields, but are suitable for tasks relative to Machine Translation, such as the ones described above. The measures are wrapped in an open source tool called RankEval which is described in detailed.

2. Previous Work

The simplest measure of its kind, tau, was introduced by Kendall (1938) with the purpose to analyze experiments on psychology, where the order given by different observers is compared. This measure has been analyzed and modified over the years for several purposes (Knight, 1966; Agresti, 1996; Christensen, 2005) and has been also applied to text technologies (Lapata, 2003; Cao et al., 2007). Since 2008 it appears modified as an official segment-level measure for the evaluation metrics in the yearly shared task for Machine Translation (Callison-Burch et al., 2008). This is the reason we decided to re-implement Kendall tau with penalization of ties, although there is already another open source version by SciPy (Oliphant, 2007), however with different accounting of ties.

More metrics emerged for use with Information Retrieval. Directed Cumulated Gain (Järvelin and Kekäläinen, 2002) was extended to the measures of Discounted Cumulative Gain, Ideal Cumulative Gain and Normalized Cumulative Gain (Wang et al., 2013). Mean Reciprocal Rank was introduced as an official evaluation metric of TREC-8 Shared Task on Question Answering (Radev et al., 2002) and has also been applied successfully for the purpose of evaluating MT n-best lists and transliteration in the frame of the yearly Named Entities Workshop (Li et al., 2009). Additionally, Expected Reciprocal Rank (Chapelle et al., 2009) was optimized for Search Engine results and used as a measure for a state-of-the-art *Learning to Rank* challenge (Chapelle and Chang, 2011).

In the following sections we present shortly the evaluation measures and the way they have been implemented to suit the evaluation needs of MT.

3. Methods

In a ranking task, each translation is assigned an integer (further called a *rank*), which indicates its quality as compared 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 aim of the methods below is to produce a score that indicates the quality of an automatically predicted ranking against human rankings.

3.1. Kendall's Tau

3.1.1. Original calculation

Kendall's tau (Kendall, 1938; Knight, 1966) 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 first decomposed into pairwise comparisons. Then, a concordant pair is counted when each predicted pairwise comparison matches the respective pairwise comparison by the human annotator; otherwise a discordant pair is counted. Consequently, tau is computed by:

$$\tau = \frac{\text{concordant} - \text{discordant}}{\text{concordant} + \text{discordant}} \tag{1}$$

with values ranging between minus one and one. The closer $|\tau|$ values get to one, the better the ranking is. In particular, when values get close to minus one, the ranking is also good, but the order of its element should be reversed. This is typical for evaluation metrics which assign higher scores to better translations, whereas humans evaluations usually assign lower ranks to the better ones. A value of zero indicates no correlation.

3.1.2. Penalization of ties

A common issue in ranking related to MT is that 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. A tie can exist in both the gold-standard ranking (as a decision by an annotator based on his judgment) and the predicted ranking (as an uncertain decision by the machine ranker).

As one can see in the fraction of equation 1, ties are not included in the original calculation of tau, which may yield improportional results when a ranker produces a huge amount of ties and only a few correct comparisons (as only the latter would be included in the denominator). Previous work includes a few tau extensions to address this issue (Degenne, 1972). We focus on the ties penalization of Callison-Burch et al. (2008) which follows these steps:

- 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 each remaining pairwise comparison, where human annotation has not resulted in a tie, every tie on the machine-predicted rankings is penalized by being counted as a discordant pair.

$$\tau = \frac{\text{concordant} - (\text{discordant} + \text{ties})}{\text{concordant} + \text{discordant} + \text{ties}} \tag{2}$$

With these modifications, the values of the ratio are still between minus one and one, but since a ties penalty has been added, values close to minus one can no longer be considered as a good result and if needed, ranks must be reverted prior to the calculation.

3.1.3. Segment-level correlation on a document level

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

- **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.[1]
- **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.

3.1.4. P-value for Kendall tau

For an amount of n ranked items, we calculate the two-sided p-value for a hypothesis test whose null hypothesis is an absence of association (Oliphant, 2007):

$$z = \frac{\tau}{\sqrt{\frac{4n+10}{9n(n-1)}}} \tag{3}$$

$$p = \mathrm{erfc}\left(\frac{|z|}{\sqrt{2}}\right) \tag{4}$$

where erfc is the complementary error function of the fraction.

3.2. First Answer Reciprocal Rank and Mean Reciprocal Rank

Kendall tau correlation sets the focus on the entire ranking list, giving an equal weight to the correct prediction of all ranks. Another set of measures emphasizes only

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

on the best item(s) (according to the humans) and how high they have been ranked by the ranker, assuming that our interest for the worse items is less. The first measure of this kind the First Answer Reciprocal Rank (FARR) which is the multiplicative inverse of the rank of the first correct answer (Radev et al., 2002), having an index i:

$$\mathrm{FARR} = \frac{1}{\mathrm{rank}_i} \tag{5}$$

A common use of FARR is through the Mean Reciprocal Rank, which averages the segment-level reciprocal ranks over all sentences:

$$\mathrm{MRR} = \frac{1}{n}\sum_{j=1}^{n}\frac{1}{\mathrm{rank}_{j,i}} \tag{6}$$

where n is the number of sentences, j the sentence index and $\mathrm{rank}_{j,i}$ the rank of the first correct answer for this sentence. As FARR is calculated over only one rank, ties need only be considered only if they occur for this particular rank. In that case, we only consider the ranker's best prediction for it.

3.3. Cumulative Gain

This family of measures is based on Discounted Cumulative Gain (DCG), which is a weighted sum of the degree of relevance of the ranked items. This introduces a *discount*, which refers to the fact that the rank scores are weighted by a decreasing function of the rank i of the item.

$$\mathrm{DCG}_p = \sum_{i=1}^{p}\frac{2^{\mathrm{rel}_i}-1}{\log_2(i+1)} \tag{7}$$

In our case, we consider that relevance of each rank (rel_i) is inversely proportional to its rank index.

The most acknowledged measure of this family is the Normalized Discounted Cumulative Gain (NDCG), which divides the DCG by the Ideal Discounted Cumulative Gain (IDCG), the maximum possible DCG until position p. Then, NDGC is defined as:

$$\mathrm{NDCG}_p = \frac{\mathrm{DCG}_p}{\mathrm{IDCG}_p}. \tag{8}$$

3.4. Expected Reciprocal Rank

The Expected Reciprocal Rank (ERR) has been suggested as an improvement of NDCG in order to better model the fact that the likelihood a user examines the translation at rank i is dependent on how satisfied the user was with the translations observed previously in the ranking list (Chapelle et al., 2009), introducing the so-called *user cascade model*.

The probability of relevance is here given by

$$R_i = \frac{2^{\mathrm{rel}_i} - 1}{2^{\mathrm{rel}_{i_{\max}}}} \tag{9}$$

and given that the user stops at position r, this forms the calculation of ERR as:

$$\mathrm{ERR} = \sum_{r=1}^{n} \frac{1}{r} \prod_{i=1}^{r=1} (1 - R_i) R_r. \tag{10}$$

3.5. Simple Measures

Additionally to the above sophisticated measures, we also use simpler measures. These are:

- **Best predicted vs human** (BPH): For each sentence, the item selected as best by the machine ranker, may have been ranked lower by the humans. This measure returns a vector of how many times the item predicted as best has fallen into each of the human ranks.
- **Average predicted**: the average human rank of the item chosen by the machine ranker as best.

3.6. Normalization of Ranking Lists

Normalization emerges as a need from the fact that in practice there are many different ways to order items within the range of the rank values. This becomes obvious if one considers ties. Since there is no standard convention for ordering ties, the same list may be represented as `[1, 2, 2, 3, 4]`, `[1, 2, 2, 4, 5]`, `[1, 3, 3, 4, 5]` or even `[1, 2.5, 2.5, 4, 5]`. The alternative representations are even more when more ties are involved.

All representations above are equivalent, since there is no absolute meaning of quality in the values involved. Nevertheless, the rank value plays a role for the calculation of some of the metrics explained above. For this purpose, we consider several different normalization options of such ranking lists:

- **minimize**: reserves only one rank position for all tied items of the same rank (e.g.: `[1, 2, 2, 3, 4]`).
- **floor**: reserves all rank positions for all tied items of the same rank, but sets their value to the minimum tied rank position (e.g: `[1, 2, 2, 4, 5]`).
- **ceiling**: reserves all rank positions for all tied items of the same rank, but sets their value to the maximum tied rank position (e.g: `[1, 3, 3, 4, 5]`). This is the default setting, inline to many previous experiments.
- **middle**: reserves all rank positions for all tied items of the same rank, but sets their value to the middle of the tied rank positions (e.g: `[1, 2.5, 2.5, 3, 4]`).

4. Implementation

4.1. Coding and Architecture

The code has been written in Python 2.7, taking advantage of the easier calculation due to the dynamic assignment of items in the lists. Few functions from `numpy` and `scipy` libraries are included, which therefore sets them as prerequisites for running the tool. The code is available in an open `git` repository.[2]

The code includes one function for each ranking measure, with the exception of NDGC and ERR which are merged into one loop for saving computational time. Each function receives as parameters the predicted and the human (also referred to as *original*) rankings. Depending on how many the results of each function are, they are returned as single float values, tuples or `dict` structures, as explained in the documentation of each function. The code is organized in two Python modules, so that the functions can be imported and used by other Python programs.

- `ranking.segment`, where the segment-level calculation takes place, and
- `ranking.set`, where the segment-level calculations are aggregated to provide results for the entire data set. This mainly includes averaging (as explained previously) but also the simple measures (Section 3.5). There is also a utility function that executes all available functions and returns the results altogether.

The ranking lists are handled by the `sentence.ranking.Ranking` class, which includes the functions for normalizing the included values.

4.2. Stand-Alone Execution

A stand-alone execution is also possible using the command line script `rankeval.py` which resides on the root of the package. This script is responsible for reading command line parameters on the execution, opening and parsing the files with the ranking lists, starting the evaluation and displaying the results. The script supports reading two formats:

- a text-based format, similar to the one used for WMT Evaluation Shared Task
- an XML-based format, which includes the sentence-level ranking annotations along with the source and translated text. This format has been used in several quality estimation tasks

4.3. Linear Computation of ERR

Since the mathematical formula for the computation of the Expected Reciprocal Rank is computed in exponential time, we use the simplified computation suggested by Chapelle et al. (2009), which is outlined in Algorithm 1. The algorithm reduces the

[2] `https://github.com/lefterav/rankeval`

Algorithm 1: Linear computation of Expected Reciprocal Rank

```
foreach $i$ in $[0, n]$ do $g_i \leftarrow$ RelevanceGrade(i)
$p \leftarrow 1, \mathsf{ERR} \leftarrow 0.$
for $r \leftarrow 1$ to $n$ do
    $R \leftarrow$ RelevanceProb($g_r$)
    $\mathsf{ERR} \leftarrow \mathsf{ERR} + p \times R/r$
    $p \leftarrow p \times (1 - R)$
return ERR
```

computational perplexity by calculating the relevance grades g_i only once for each rank i. This is used during the loop for calculating the relevance probability R_i and gradually augmenting the ERR value.

5. Discussion

It is hard to evaluate new metrics, as we examine a meta-evaluation level, where there is no gold standard to compare with. Therefore, we leave this kind of evaluation to further work, as we hope that the tool will make it possible to apply the measures on different types of data.

As an indication of the correlation between the measures in a range of experiments we present a graphical representation (Figure 1) of all measure values, given for 78 quality estimation experiments on ranking. These experiments were done with various machine learning parametrizations (Avramidis, 2012) over the basic set-up of the Sentence Ranking Shared Task on Quality estimation (WMT13 – Bojar et al., 2013). The experiments are ordered based on their descending MRR score, which appears as a straight line, whereas the scores given by the other measures for the respective experiments are plotted with the rest of the lines.

Each measure has a different range of values, which means that the position on the Y axis, or the inclination are of no interest for the comparison. The interesting point are the fluctuations of each measure scores as compared to the others. As expected, we see that the measures of the same family seem to correlate with each other.

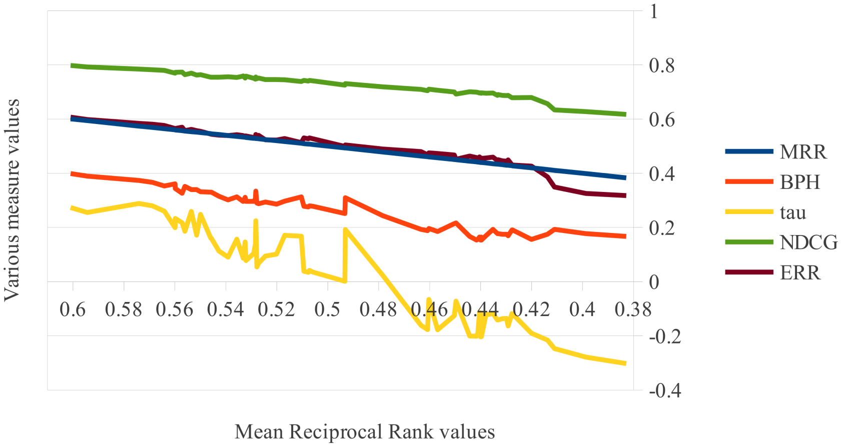

Figure 1. Plotting the values of the various measures (Y axis) for 78 quality estimation experiments ordered by descending MRR (X axis)

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 Popović and Dr. David Vilar for their useful feedback.

Bibliography

Agresti, Alan. *An introduction to categorical data analysis*, volume 135. Wiley New York, 1996.

Avramidis, Eleftherios. Comparative quality estimation: Automatic sentence-level ranking of multiple machine translation outputs. In *Proceedings of 24th International Conference on Computational Linguistics*, pages 115–132, Mumbai, India, Dec. 2012. The COLING 2012 Organizing Committee.

Bojar, Ondřej, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. Findings of the 2013 workshop on statistical machine translation. In *8th Workshop on Statistical Machine Translation*, Sofia, Bulgaria, 2013. Association for Computational Linguistics.

Callison-Burch, Chris, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. Further meta-evaluation of machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 70–106, Columbus, Ohio, June 2008. Association for Computational Linguistics.

Cao, Zhe, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136. ACM, 2007.

Chapelle, Olivier and Yi Chang. Yahoo! learning to rank challenge overview. *Journal of Machine Learning Research-Proceedings Track*, 14:1–24, 2011.

Chapelle, Olivier, Donald Metlzer, Ya Zhang, and Pierre Grinspan. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM conference on Information and knowledge management - CIKM '09*, page 621, New York, New York, USA, Nov. 2009. ACM Press. ISBN 9781605585123. doi: 10.1145/1645953.1646033.

Christensen, David. Fast algorithms for the calculation of Kendall's τ. *Computational Statistics*, 20(1):51–62, 2005.

Degenne, Alain. *Techniques ordinales en analyse des donn{é}es statistique*. Classiques Hachette, 1972.

Järvelin, Kalervo and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, Oct. 2002. ISSN 10468188. doi: 10.1145/582415.582418.

Kendall, Maurice G. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93, 1938. doi: 10.1093/biomet/30.1-2.81.

Knight, William R. A computer method for calculating kendalls tau with ungrouped data. *Journal of the American Statistical Association*, 61(314):436–439, 1966.

Lapata, Mirella. Probabilistic text structuring: Experiments with sentence ordering. In *Annual Meeting of the Association for Computational Linguistics*, pages 545–552, 2003.

Li, Haizhou, A Kumaran, Vladimir Pervouchine, and Min Zhang. Report of NEWS 2009 machine transliteration shared task. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)*, pages 1–18, Suntec, Singapore, Aug. 2009. Association for Computational Linguistics.

Oliphant, Travis E. SciPy: Open source scientific tools for Python. *Computing in Science and Engineering*, 9(3):10–20, 2007. URL `http://www.scipy.org`.

Radev, Dragomir, Hong Qi, Harris Wu, and Weiguo Fan. Evaluating web-based question answering systems. In *Proceedings of the Third International Conference on Language Resources and Evaluation*, volume 1001, Las Palmas, Spain, 2002. European Language Resources Association (ELRA).

Wang, Yining, Wang Liwei, Yuanzhi Li, Di He, Wei Chen, and Tie-Yan Liu. A theoretical analysis of NDCG ranking measures. In *26th Annual Conference on Learning Theory*, 2013.

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