πRate: A Task-oriented Monolingual Quality Estimation System

Zijian Győző Yang[1] and László János Laki[2,3]

[1] Pázmány Péter Catholic University, Faculty of Information Technology and Bionics
[2] MTA-PPKE Hungarian Language Technology Research Group
Práter str. 50/A, 1083 Budapest, Hungary
[3] MorphoLogic Lokalizci Kft.
Logodi str. 54, 1012 Budapest, Hungary
{yang.zijian.gyozo, laki.laszlo}@itk.ppke.hu

Abstract. Psycholinguistically motivated natural parsing is a new, human-oriented computational language processing approach. This complex real-time model has several parallel threads to analyze the input words, phrases or sentences. One of the main threads can be the quality estimation module, which informs, controls and filters the noisy or erroneous input. To build this quality controller module we implemented the quality estimation method that is traditionally used in the field of machine translation evaluation. To tailor the quality estimation model to the monolingual natural parsing system, we optimized the architecture with task-oriented approach. In our research, a quality estimation system is built for monolingual text input. Using this system we can provide quality indicators for the input with an accuracy of $\sim 72\%$. The system is created for the `AnaGramma` Hungarian natural parsing system, but it can be used for other languages as well. Our method can incrementally estimate the quality of input in real time while it is generated.

1 Introduction

Psycholinguistics has become a very important field in speech and language processing. While most of the traditional linguistics parsers (syntactic parsers, part-of-speech taggers etc.) start analysis after the end of a sentence has been given, in human conversation, people process sentences word-by-word immediately after they have read or heard the input words or utterances. Therefore, a new approach is needed, which simulates the real, strictly left-to-right human language processing.

`AnaGramma` [4, 12] is a psycholinguistically motivated monolingual natural parsing system, which models real human language processing. `AnaGramma` is a complex real-time system, which has several parallel threads to analyze input words and utterances. Besides the morphological analyzer, corpus statistics and other threads, there is another important main thread: the quality analyzer thread.

The main task of the quality analyzer is to provide reliable quality indicators for the other threads and the reader, or in the particular case, control or filter the

input. The `AnaGramma` parser cannot handle or is not responsible for handling non-Hungarian or totally wrong sentences. A quality control system, however, can filter out the problematic segments. Furthermore, `AnaGramma` has two basic types of threads: *offer* thread, which provides information about the current element (e.g. the particular element is in nominative case), and *demand* thread, which is looking for required elements with specific properties (e.g. a transitive verb needs its object). During the processes of the offer and the demand threads, there are too many possible offers that will be collected by the demands, which include a lot of irrelevant offers. A quality control system with specific features could be able to filter them. Last but not least, if there are several correct offer elements for a demand thread, a quality estimation score can suggest to select the better offer. Thus, different threads need different quality indicators and the system or the reader needs local and global quality scores of the input sentences. Hence it is important that the quality control system should be flexible, schedulable and the quality features have to be clearly separated in terms of the given task. Thus the standard quality estimation (QE) systems like QuEst [15], could not perform well according to our requirements.

The structure of this paper is as follows: After a short review of related approaches, we will present the details of our new QE system, which is called πRate system. Then, we will present our experiments, optimizations and results. Finally, conclusions and future plans are described.

2 Related Work

Traditional QE is a prediction task (see Figure 1), where different quality indicators are extracted from the source and the machine translated segments. The QE model is built with machine learning algorithms based on these quality indicators. Then, the QE model is used to predict the quality of unseen translations. The aim is that the scores, predicted with the QE model, highly correlate with human judgments. Thus the QE model is trained on human evaluated sentence pairs.

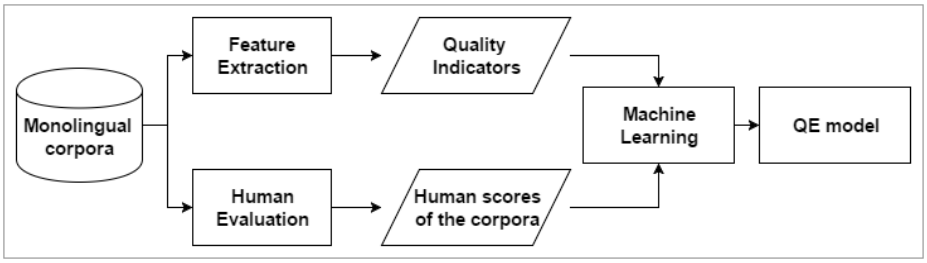

Fig. 1. The QE model

In our research, we have only one language, thus we used monolingual corpora and features to train the QE model.

The QuEst++ [14] system has word level QE, which contains monolingual evaluations, such as LM features, syntactic features, target context features, etc. But it is only a part of the whole translation evaluation system and it is not a specifically monolingual QE tool.

There are various research or business fields using task-oriented or service-oriented architectures. Such fields are for example, in e-commerce [8], robotics [10], automated video surveillance [6], etc. The advantage of the task-oriented architecture is that it separates the model into tasks. A task is an independent unit of functionality. Different tasks may use different resources and according to the various types of tasks, we can schedule them. With effectively organized scheduling, we can flexibly optimize the performance and serve the specific needs. We used the task-oriented approach to the QE method.

3 πRate system

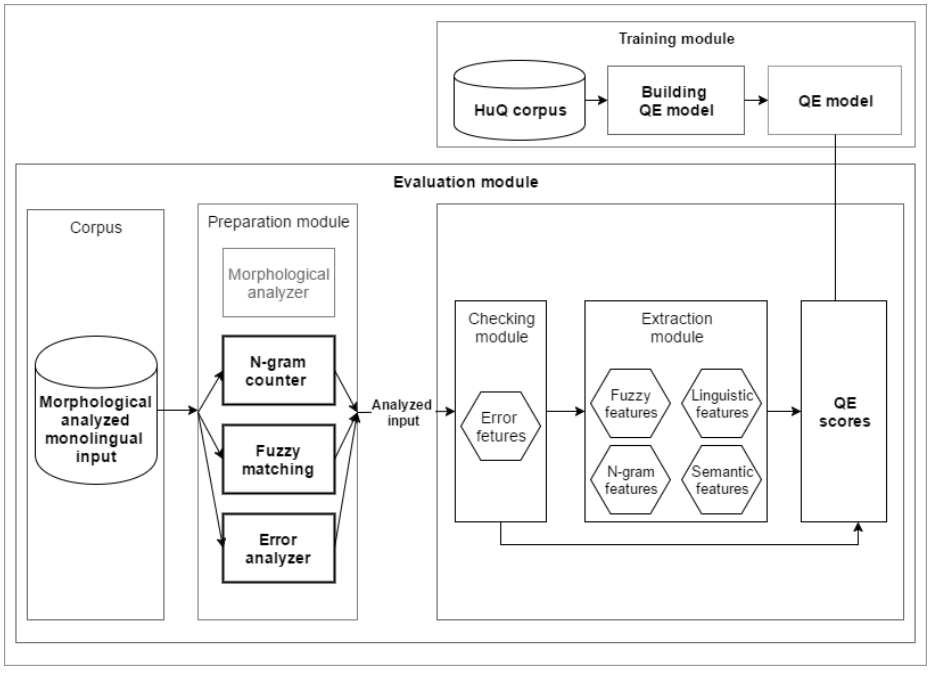

Fig. 2. The πRate system

We implemented a monolingual QE system, which is called πRate[4] system. There are two main modules in the πRate system (see Figure 2): training module and evaluation module.

3.1 Training module

The main function of the training module is to build the QE model. This module is the same as the standard QE training model (see Figure 1). In this training phase, we use a monolingual corpus to build the QE model. The training corpus is evaluated by human annotators.

There is another main part of the training model: feature extraction. We have different kinds of monolingual features: linguistic features, n-gram features, fuzzy features, error features, etc. Using these features, quality indicators are extracted from the training corpus. Then, using these quality indicators and the human evaluation scores, the QE models are built.

3.2 Evaluation module

In the case of `AnaGramma`, the input can be morphologically analyzed, thus according to the condition of input, there are features that can be skipped (see in Figure 2, the morphological analyzer is grey). This task-oriented scheduling is one of the advantages of the πRate system. The separation of the tasks can optimize the resources and the performance.

The evaluation module has three main parts:

- preparation module;
- checking module;
- extraction module.

The input text incrementally increases. There are different kinds of thread analyze the input, hence the πRate system may get a morphologically analyzed input. After the input words arrive in the system, the preparation module analyzes the input. It counts n-gram probabilities, perplexities, fuzzy matching, etc. Then, it forwards the analyzed input to the feature extraction module. First, the error features are extracted. Then, the checking module carries out controls. If the error values achieve an error threshold, the checking module warnings the reader or filters out the input, otherwise it allows the remaining features to extract the quality indicators. After the feature extraction, the πRate system estimates the special, the local and the global quality scores. The special quality scores are counted for other threads, the local quality score is the quality of the current sentence or utterance. The global quality is the quality of the whole input text till the current word.

[4] Our research group is in room 314.

4 Methods and Experiments

For building the QE model, monolingual features as quality indicators are needed, which are extracted from the monolingual corpus. Then, machine learning methods and human evaluation scores are used to build the QE model (see Figure 1). To build the πRate system we used JAVA EE, and REST architecture.

4.1 HuQ corpus

For training, we used the HuQ corpus [17]. The HuQ corpus contains 1500 Hungarian sentences. All the 1500 sentences were evaluated by 3 human annotators. All the annotators were native Hungarian speakers. For evaluation, they used the Likert scoring scale from 1-5. For building the QE models, we used the arithmetic mean of the scores of the annotators. The HuQ corpus also has classification scores, because there are many cases, when we do not need 5 grades. There are 3 classes were created from the human scores: BAD: 1 <= score <=2; MEDIUM: 2 < score < 4; GOOD: 4 <= score <=5. There are 3 subcorpora in the HuQ corpus: subtitle, literature and law.

For our experiments, we divided the HuQ corpus into 500-1000 sentences for fuzzy reference and QE model.

The 500 fuzzy reference sentences was custom selected. There are mostly equals parts of "BAD", "MEDIUM" and "GOOD" classes (167 BAD sentences, 166 MEDIUM sentences, 167 GOOD sentences).

The 1000 sentences for the QE model were separated in a ratio of 90%-10% for training and test sets. For testing, we used 10-fold cross validation.

4.2 Monolingual features

In our experiment we have different kinds of monolingual features. According to the functionality, we can separate the features into the following categories:

- linguistics features:
  - percentage of nouns, verbs, pronouns, adverbs, adjectives, conjunctions, determiners, preverbs, interjections in the sentence;
  - ratio of number of nouns and verbs in the sentence;
  - ratio of number of nouns and adjectives in the sentence;
  - ratio of number of verbs and preverbs in the sentence;
  - ratio of number of nouns and determines in the sentence;
- n-gram features:
  - sentence LM probability;
  - sentence LM perplexity;
  - sentence LM perplexity without end of sentence marker;
  - LM probability of lemmas, POS tags of the sentence;
  - LM perplexity of lemmas, POS tags of the sentence;
  - LM perplexity of lemmas without end of sentence marker;
- fuzzy features:

    - Inspired by the semantic similarity features, developed by Hanna Bechara et al. [1], we used 1/3 of the HuQ corpus as a reference. In this reference corpus, using fuzzy matching we find the most similar sentence to the input, then we extract the human evaluation scores (likert and classification scores) of the matched sentence. For fuzzy matching we used Levenstein distance, BLEU [9], NIST [5], TER metrics and semantic similarity methods: LSI and word embedding. There are many cases, when the fuzzy search finds more similar sentences for the input. We used the semantic similarity methods to filter the result of the searches.
- error features:
    - percentage of xml tags in the sentence;
    - percentage of non-English words in the sentence;
    - percentage of unknowns words in the sentence;
    - percentage of punctuation marks in the sentence;

Using the monolingual features and the HuQ corpus we built the QE models:

- LS model: QE model using the Likert scores.
- CS model: QE model using the Classification scores.

Inspired by the research of quality estimation for Hungarian [18], to train the QE models, we used the SVR (support vector regression) and the SVM (support vector machine).

After the QE models were built, we implemented the πRate system. First, the preparation module works. For Part-of-Speech (POS) tagging and lemmatization, we used PurePos 2.0 [7], which is an open source, HMM-based morphological disambiguation tool. Purepos2 has the state-of-the-art performance for Hungarian. It has the possibility to integrate a morphological analyzer. Thus, to get the best performance, we used Humor [11], a Hungarian morphological analyzer. For NP-chunking, we used HunTag [13] that was trained on the Szeged Treebank [3]. HunTag is a maximum entropy Markov-model based sequential tagger. For n-gram counting, we used the SRILM Toolkit [16]. To find fuzzy matching, we used the segment level BLEU, segment level NIST, TER and Levenstein distance metrics. For semantic matching we counted the LSI and the word embedding vectors from the input text.

During the features extraction phase, first, the checking module controlled the input. The error features filtered out the sentences that achieved the error threshold. Then, using the QE model, the current quality scores are counted. At the end, the πRate system inform the reader and the other threads about quality status.

We also performed the optimization task. According to machine translation evaluation [2], not all the features are relevant to the QE model. We used the forward selection method [18] to find the optimized feature sets:

- OptLS set: Optimized feature set for LS model.
- OptCS set: Optimized feature set for CS model.

5 Results and Evaluation

For evaluating the performance of πRate system, we used the statistical correlation, the Mean absolute error (MAE), the Root mean-squared error (RMSE) and the Correctly Classified Instances (CCI) evaluation metrics. The correlation ranges from -1 to +1, and the closer the correlation to -1 or +1, the better it is. In the case of MAE and RMSE, the closer the value to 0, the better.

Using the HuQ corpus and 32 features we built the QE models and the πRate system. In Table 1 and in Table 2, we can see that the 32 feature set could gain $\sim$59% correlation, and $\sim$70% correctly classified instance results.

	Correlation	MAE	RMSE
LS model - 32 features	0,5936	0,6857	0,8961
OptLS set - 13 features	0,6278	0,6783	0,8758

Table 1. Evaluation of LS model and OptLS set

	CCI	MAE	RMSE
Cs model - 32 features	70,7%	0,2465	0,3590
OptCS set - 8 features	71,7%	0,2544	0,3539

Table 2. Evaluation of CS model and OptCS set

We did optimization with forward selection method. After optimization, as we can see in Table 1 and in Table 2:

- The OptLS set, using only 13 features, could gain $\sim$3% higher correlation.
- The OptCS set, using only 8 features, could gain $\sim$1% more correctly classified instances.

In Table 3 and in Table 4 we can see the optimized feature sets (sorted by the degree of the improvement of the models). From the fuzzy features, there is no LSI features. There are only word embedding features, which means the word embedding method is better than the LSI. The most significant feature in both models is the LM (n-gram) probability. This means that in Hungarian, the word order and the sentence structure are important aspects.

In Table 5, we can see correct and incorrect quality predictions. In the cases of incorrect prediction, there are samples where the Likert model predicted correctly and the Classification model did not, and vice versa, but there are samples where none of them estimated correctly. The explanation for this is that, in most cases, the fuzzy feature and LM probability features caused the difference between the actual and the predicted scores. For instance, in the last sample, the

Feature
LM probability of POS tags of the sentence
Conjunctions / sentence length
Fuzzy matching - word embedding model (Likert score)
LM probability of lemmas of the sentence
Nouns / sentence length
NIST fuzzy matching (using word embedding model) - Classification score
LM perplexity of POS tags of the sentence
Adjectives / sentence length
Punctuation marks / sentence length
Verbs / preverbs
Preverbs / sentence length
Unknown words / sentence length
Fuzzy matching - word embedding model (Classification score)

Table 3. Optimized 13 features for the Likert QE model

Feature
LM probability of the sentence
Sentence LM perplexity
Conjunctions / sentence length
TER fuzzy matching (using word embedding model) - Classification score
Levenstein fuzzy matching (using word embedding model) - Likert score
Sentence LM perplexity without end of sentence marker
LM perplexity of lemmas of the sentence
Punctuation marks / sentence length

Table 4. Optimized 8 features for the Classification QE model

fuzzy matching feature for the "Ők soha csinál amit."('They never does what.') sentence matched the "Ők soha nem az." ('They never that.') sentence with actual scores: Likert - 3.333; Classification - MEDIUM. The remaining features further improved the scores.

Likert score		Classification score		Sentence
Actual	Predicted	Actual	Predicted	
4.333	4.559	GOOD	GOOD	Mahmoud eltorzította az arcát. (Mahmoud contorted his face.)
3.667	3.198	MEDIUM	MEDIUM	Megyek az öltönyt. (I am going the suit.)
2	1.683	BAD	BAD	Az elnök a magát a vége felé, a nebraskai. (The president the himself towards the end, the Nebraskan.)
2	3.531	BAD	BAD	A többi súlyos szó, és hidrokarbon létfontossg. (The other heavy words and hydrocarbon are vital.)
5	2.520	GOOD	GOOD	Senki sem tudja. (Nobody knows.)
2	3.628	BAD	GOOD	Ők soha csinál amit. (They never does what.)

Table 5. Samples for correct and incorrect quality estimation

6 Conclusion

We created the πRate task-oriented quality estimation system for monolingual natural parsing. Standard QE models do not perform well enough for this task, therefore we modified and optimized the architecture, that is inspired by the task-oriented architectures. Using this task-oriented architecture, we could schedule the features and modules flexible to fit the different types of inputs and tasks. To build the QE models and the evaluation system, we implemented the πRate system. To build the system, we used the HuQ corpus and 32 features. We did feature optimization tasks as well. After optimization, with less features, we could gain $\sim$3% higher correlation result and $\sim$1% more correctly classified instances. The πRate system can produce $\sim$72% accuracy estimation for Hungarian. The πRate system is created for the `AnaGramma` natural parsing system, but it can be used for other languages as well. Our method can incrementally estimate the quality of input in real time while it is generated.

7 Future plans

The πRate system can be the state-of-art QE system for monolingual natural parsing. We implemented this system for Hungarian, but it can be used for other monolingual natural parsing systems. Our future plan is investigating more features. We also would like to improve the fuzzy matching features and the reference corpus.

References

1. BECHARA, H., ESCARTIN, C.P., ORASAN, C., SPECIA, L.: Semantic textual similarity in quality estimation. Baltic Journal of Modern Computing, Vol. 4 (2016), No. 2 pp. 256–268 (2016)
2. Beck, D., Shah, K., Cohn, T., Specia, L.: Shef-lite: When less is more for translation quality estimation. In: Proceedings of the Workshop on Machine Translation (WMT) (2013)
3. Csendes, D., Csirik, J., Gyimóthy, T., Kocsor, A.: The Szeged Treebank. In: Lecture Notes in Computer Science: Text, Speech and Dialogue. pp. 123–131 (2005)
4. Indig, B., Laki, L., Prószéky, G.: Mozaik nyelvmodell az anagramma elemzhz. In: XII. Magyar Szmtgpes Nyelvszeti Konferencia. pp. 260–269. Szegedi Tudományegyetem, Informatikai Tanszékcsoport, Szeged, Hungary (2016)
5. Lin, C.Y., Och, F.J.: Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In: Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics. ACL '04, Association for Computational Linguistics, Stroudsburg, PA, USA (2004), http://dx.doi.org/10.3115/1218955.1219032
6. Monari, E., Voth, S., Kroschel, K.: An object- and task-oriented architecture for automated video surveillance in distributed sensor networks. In: Proceedings of the 2008 IEEE Fifth International Conference on Advanced Video and Signal Based Surveillance. pp. 339–346. AVSS '08, IEEE Computer Society, Washington, DC, USA (2008), http://dx.doi.org/10.1109/AVSS.2008.21
7. Orosz, G., Novák, A.: Purepos 2.0: a hybrid tool for morphological disambiguation. In: RANLP'13. pp. 539–545 (2013)
8. Papazoglou, M.P., Heuvel, W.J.: Service oriented architectures: Approaches, technologies and research issues. The VLDB Journal 16(3), 389–415 (Jul 2007), http://dx.doi.org/10.1007/s00778-007-0044-3
9. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: A method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. pp. 311–318. ACL '02, Association for Computational Linguistics, Stroudsburg, PA, USA (2002), http://dx.doi.org/10.3115/1073083.1073135
10. Parker, L.E.: Task-oriented multi-robot learning in behavior-based systems. In: Intelligent Robots and Systems '96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on. vol. 3, pp. 1478–1487 vol.3 (Nov 1996)
11. Prószéky, G.: Industrial applications of unification morphology. In: Proceedings of the Fourth Conference on ANLP. pp. 213–214. Stuttgart, Germany (1994), http://www.aclweb.org/anthology/A94-1046

12. Prószéky, G., Indig, B.: Natural parsing: a psycholinguistically motivated computational language processing model. In: 4th International Conference on the Theory and Practice of Natural Computing. Mieres, Spain (2015)
13. Recski, G., Varga, D.: A Hungarian NP Chunker. The Odd Yearbook. ELTE SEAS Undergraduate Papers in Linguistics pp. 87–93 (2009)
14. Specia, L., Paetzold, G., Scarton, C.: Multi-level translation quality prediction with quest++. In: ACL-IJCNLP 2015 System Demonstrations. pp. 115–120. Beijing, China (2015), http://www.aclweb.org/anthology/P15-4020
15. Specia, L., Shah, K., de Souza, J.G., Cohn, T.: Quest - a translation quality estimation framework. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations. pp. 79–84. Sofia, Bulgaria (2013), http://www.aclweb.org/anthology/P13-4014
16. Stolcke, A.: Srilm - an extensible language modeling toolkit. pp. 901–904 (2002)
17. Yang, Z.G., Laki, J.L., Siklósi, B.: HuQ: An english-hungarian corpus for quality estimation. In: Proceedings of the LREC 2016 Workshop - Translation Evaluation: From Fragmented Tools and Data Sets to an Integrated Ecosystem
18. Yang, Z.G., Laki, L.J., Siklsi, B.: Quality estimation for english-hungarian with optimized semantic features. In: Computational Linguistics and Intelligent Text Processing. Konya, Turkey (2016)