Phrase-level Quality Estimation for Machine Translation

Varvara Logacheva, Lucia Specia

Department of Computer Science
University of Sheffield, UK
v.logacheva@sheffield.ac.uk, l.specia@sheffield.ac.uk

Abstract

The paper presents the first attempt to perform quality estimation (QE) of machine translation (MT) at the level of phrases. Automatically translated sentences directly or indirectly labelled by humans for quality at the word level are used to devise phrase-level quality labels. We suggest methods of segmenting sentences into phrases which mimic the actual segmentation that generated the translations. For the prediction models, we apply two sets of phrase-level features: (1) features used in sentence-level QE work, (2) features based on word vector representations. Our experiments show that the phrase-level models can improve over word-level models in terms of how well they detect errors.

1. Introduction

Quality estimation (QE) of machine translation (MT) aims at determining the quality of an automatically translated text without comparing it to a reference translation. This task often arises in real-world applications of MT, e.g. when users of an MT system translate new data and are interested in understanding how reliable the system output is. No reference translations are available for such data, and therefore the use of standard MT evaluation metrics is not possible. The only way of determining the quality of the automatic translation is the use of indirect evidence. QE is particularly useful in applications which provide automatic translations for gisting and in computer-assisted translation (CAT) settings where automatic translation is followed by post-editing by humans.

The QE task started as the estimation of confidence of individual words in a translated sentence with respect to a particular translation model. Back then the task focused on the confidence of a particular MT system about an automatic translation, and as such explored features that required information from the MT system, such as hypotheses and n-best lists statistics [1], word posterior probabilities [2], n-gram posterior probabilities [3].

More recently QE acquired a broader sense [4]: estimating the quality of a translation for a particular purpose (e.g. gisting or further post-editing), often disregarding the MT system that generated it. The features currently used in QE are thus system-independent; they use properties of the source text and its translation (e.g. number of tokens, numbers, punctuation marks in sentences) or information from external resources not related to the MT system that produced the translation (POS tags, syntactic features, perplexity under external LMs) [5].

The labelling of translations (and therefore the score to estimate) has changed as well: instead of using automatic MT evaluation metrics to produce labels, the labelling is more often done by humans (e.g. post-editing effort of a sentence within to a 1-5 point scale [4]) or deduced from manually generated data (e.g. post-editing effort defined by the percentage of editing a translator performed, or post-editing time measured by a CAT tool [6]). These are all labels for sentence-level QE. Word-level labels, on the other hand, are less clearly defined.

The task of word-level QE has regained attention since 2013, when it became part of the WMT evaluation campaign [6]. The post-editing of MT output was used to automatically collect translations annotated for quality at the word level: a word left unchanged by a translator was labelled as "OK", while a word edited was labelled as "BAD". However, framing the QE task in this way has serious limitations. Notably, the fact that errors in different words are not independent from one another. For example, if two words agree in their grammatical features, changing one of them will most likely cause the need to change the other one as well. For example, if we translate the English phrase "My dear friend" into French, a possible translation is "Mon cher ami". However, a post-editor will change it into "Ma chère amie" if "friend" refers to a feminine entity. Here one mistranslation ("ami" instead of "amie") will have resulted in three corrections.

Such groups of related edits were defined in [7] as post-editing actions (PEAs) — minimal units that should be post-edited jointly in one action according to some pattern. The MQM (Multidimensional Quality Metrics) framework [8] for translation error analysis also focuses on defining errors that can span phrases of any length. This leads us to the idea that QE should be done at the level of phrases, as opposed to words. Analysing groups of words jointly can provide additional information which is not available at the word level, and notifying a user that the errors in several adjacent words are related can help them use quality predictions more efficiently.

Another motivation for phrase-level QE is the fact that the most widely used MT engines are phrase-based, i.e. at each step the MT decoder extends the translation hypothe-

sis with a phrase. In other words, decisions are made over phrases, rather than over single words. Therefore, it is likely that translation errors can also be generated at the phrase-level. In addition, phrase-level QE models could be used to guide decoding to avoid certain errors.

Previous work on word-level QE has highlighted the intuition that errors can span over entire phrases. [9] use a number of features that rely on the source phrase that generated the current target phrase. In [2] the word posterior probability is computed at the phrase level: it is regarded as the probability of a word being generated by a source phrase rather than by the entire source sentence. However, in previous research the quality labels are defined for every word, and thus our work represents the first effort to estimate the quality of a target phrase as an atomic unit. We identify the main challenges in this task and suggest ways of dealing with them.

The biggest challenge in phrase-level QE is segmentation: the task requires the automatic translations to be segmented into phrases, and each phrase to be labelled for quality. Although there exist datasets labelled for errors at the phrase level (e.g. using the MQM framework [10]), they do not provide a segmentation that can be used directly for the task. Since only errors are labelled, very long sequences of error-free segments are found in these datasets, and there is no clear way to segment them. If we train a classifier based on such data to discriminate between good and bad phrases, it is very likely to be biased by a phrase length and to classify shorter phrases as bad and longer phrases as good regardless of their actual quality. In addition, if the phrase segmentation is done based on the reference labels, we have no way of segmenting unseen data, for example the test data to evaluate the model's performance.

Since no existing phrase-labelled datasets can be used for the task, we explore and adapt datasets labelled for quality at the word level. We expand this labelling by performing decoder-like segmentation. We test different sets of features and compare the performance of phrase-level QE models on different feature sets.

The rest of the paper is organised as follows. In Section 2 we describe our segmentation strategies and ways of adapting word-level labels for phrases. Sections 3 and 4 describe the feature sets and training algorithms and in Section 5 we report the results of our experiments.

2. Segmentation and labelling

Phrase-level QE relies heavily upon appropriate sentence segmentation. One of the main difficulties involved in the segmentation task is the lack of a strict definition of what a *phrase* is for this purpose. In linguistics, *phrase* is a unit where words are connected by dependency relationships. In statistical MT, phrases are simply sequences of words that frequently co-occur and are aligned with the same source word sequences.

Given that a lot of the translation data is likely to be produced by statistical MT systems nowadays, for this work we assume the latter notion of segmentation and reproduce the segmentation produced by a statistical MT decoder. Since we do not have access to the MT system that produced the translations, we re-decode the source data with a statistical MT system and reproduce its phrase segmentation. We are not guaranteed that this segmentation will match the original one, i.e., the one that generated the target data. However, the two MT systems are very similar, and thus we hope to get similar segments. We suggest two ways of segmenting sentences into Moses-like phrases [11]: segmentation of both source and target sentences jointly with a source-target MT system, and independent segmentation of target sentences.

2.1. Source segmentation

The datasets we use for QE systems training have source sentences and their automatic translations. If we had access to the MT system which generated the translation, we could reproduce the original segmentation accurately by simply re-decoding the source sentences. However, such MT models are rarely made available, and we are not guaranteed to get the same output using another MT system, even if it trained on the same data.

One possible solution is to constrain the decoder to use only phrases that appear in the target sentence. However, constrained decoding is often unable to fully reach the translation provided, usually because of out-of-vocabulary (OOV) words or lack of suitable phrases in the phrase table. In order to supply the system with this information we trained an additional phrase table on the data to be decoded (i.e. phrase-level QE data), and produced translations using both phrase tables.

Despite this additional data-specific phrase table, a small percentage of sentences still could not be decoded. In those cases we considered each word of the sentence a separate phrase, and the corresponding source phrase as the word aligned to it. Therefore, for some "phrases" of such sentences, the source phrase will be empty.

2.2. Target segmentation

Our second technique consists in segmenting only the target sentence with an MT system which translates from the target language into the source language. We translate the target sentence with no constraints and retrieve the phrase segmentation for it. The actual translation will not match the source side of our data, which is not an issue as we will not use it. Moreover, we suppose that the output language of such a system is not important, because we only use it to segment the input sentence.

We obtain the source segmentation by combining the target segmentation and source-target alignments: for each target phrase, the corresponding source phrase is composed of all source words aligned to the words in the target phrase. The source phrase needs to be continuous, i.e. if two source

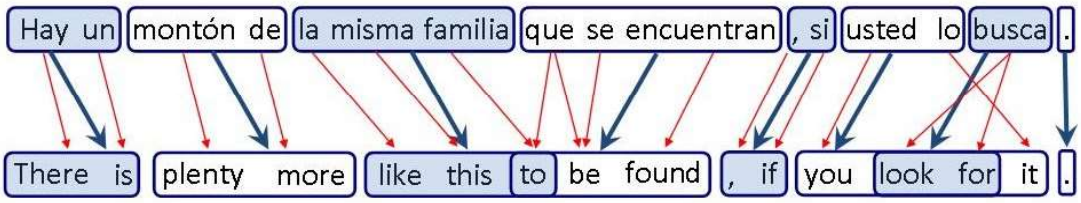

Figure 1: Overlapping source phrases generated by projection of target phrases onto the source sentence. Red lines denote word alignments, blue lines denote phrase alignments.

words aligned to one target phrase have an unaligned word (or a word aligned to another target phrase) in between, this unaligned word also has to be included into the corresponding target phrase. Figure 1 shows this case: the phrase *"usted lo"* is aligned to two source words: *"you"* and *"it"*. They have two words between them, so the source phrase will be *"you look for it"*. The figure also gives an example of overlapping source phrases.

This approach has a few drawbacks: it can include a source word in more than one phrase, it does not guarantee the full coverage of a source sentence and it can generate an empty source phrase if all words in the target phrase are unaligned. In addition to that, when performing this type of segmentation we feed the automatically translated sentences to the decoder, because the target side of our QE training data has been generated by an MT system. Since the majority of these sentences contain errors, the phrase segmentation for them can be different from the one generated for a valid target language sentence.

2.3. Phrase labelling

Datasets with post-edited machine translations can be labelled at the word level by comparing the automatic translations with its post-edited version. This can be done with edit distance metrics such as the one implemented in the Tercom tool [12]. This tool identifies an edit operation (substitution, deletion, shift) which needs to be performed on a word to make the automatic translation match its post-edition. The word labels could thus be the edit operations which need to be performed on words to improve the sentence translation, as in the dataset created for the WMT-13 QE shared task [6]. Other datasets have incorrect words manually labelled with fine-grained error classes (grammatical error, mistranslation, etc.) [10]. However, since the number of errors is relatively small (10-30% for different datasets), in order to reduce sparsity, binary ("OK"/"BAD") labels are often used [10, 13]. They indicate simply whether a word suits the context or needs to be edited. However, both these types of labels are defined over words only. When segmenting a sentence with one of the techniques described above, we are likely to face a situation where words put together into a phrase have different tags. Thus we need to combine word labels to get to a single phrase label.

The most obvious combination strategy is *majority labelling*, i.e. to assign the most common label of the words in the phrase to that phrase. However, such a strategy is likely to further increase the skewed discrepancies between the number of occurrences of "BAD" and "OK" labels. The majority tagging strategy can reduce even more the number of "BAD" tags, which will in turn make learning harder. We propose three alternative labelling strategies to mitigate this issue:

- optimistic — if half or more of words have a label "OK", the phrase has the label "OK" (majority tagging),
- pessimistic — if 30% words or more have a label "BAD", the phrase has the label "BAD",
- super-pessimistic — if any word in the phrase has a label "BAD", the whole phrase has the label "BAD".

The latter strategy is motivated by the possibility of using phrase-level QE to support phrase-based MT decoding. At each step of the search process the decoder chooses a new phrase, and the best candidate phrase should contain only "good" words. If one of the words does not fit into the context, the entire phrase should be considered unsuitable.

2.4. Joint target+data segmentation

Instead of changing the edit distance-based labels, we can get rid of phrases with ambiguous tags if we combine the phrase borders identified by the decoder with the borders of "OK" and "BAD" spans in our data. Let us consider the following example. The target phrase *"¿Sabes lo que voy a hacer, sin embargo?"* and its original edit distance-based tagging *"OK OK OK OK BAD BAD BAD BAD BAD BAD OK"* create the following segmentation:

[¿ Sabes lo que] [voy a hacer , sin embargo] [?]

The **target segmentation** procedure for the same sentence returns a different segmentation with ambiguous tags:

[¿ Sabes] [lo que voy a hacer] [,] [sin embargo] [?]

OK OK/BAD BAD BAD OK

However, if we combine two sets of borders, we convert one phrase with ambiguous tagging (*"lo que voy a hacer"* — 2 "OK", 3 "BAD" words) into two unambiguous phrases:

[¿ Sabes] [lo que] [voy a hacer] [,] [sin embargo] [?]

OK OK BAD BAD BAD OK

Note that we can join the phrase borders with the label span borders only for the target segmentation, because the source segmentation has the corresponding source phrases, which cannot be segmented into "BAD" and "OK" segments.

2.5. Evaluation

In order to implement a phrase-level QE system we need to segment both training and test data, and then label the test phrases with a trained model. However, the phrase-level output currently cannot be evaluated directly, because we have no datasets with phrase-level annotation. Therefore, we segment the test sentences into phrases and label them, and then we propagate the phrase labels onto all words of the phrase. After that the test output can be evaluated at the word level.

3. Features

The the majority of features used in word-level QE systems cannot be applied to phrases. However, most of the sentence-level features are suitable for any sequence of words, not only full sentences. For our experiments we used a list of 79 sentence features used in the QuEst QE framework [14][1]. These features are called "black-box" because they do not use the information from MT system. Some examples:

- LM features: language models (LM) score of source and target phrases under source and target LMs.
- POS features: numbers of verbs, nouns and other parts of speech in the source and the target.
- Features that indicate the number of tokens from different closed classes: numbers, alphanumeric tokens, punctuation marks.
- Average number of translations of source words.
- Average number of n-grams in different frequency quartiles.

Another set of features we use relies on source-only information, namely vector representations of words generated with `word2vec` tool[2]. Word2vec assigns every word a fixed-size vector of numbers that encodes information on the word's contexts. Therefore, similar words should have similar vectors (for a detailed description of `word2vec` see [15]). The vectors are word-level, but unlike other word-level features they can be easily combined for phrases that are longer than one word. We can use two vector operations to combine two or more vectors of the same size while keeping the dimensionality of these vectors: element-wise sum or average of the vectors. According to our preliminary experiments, systems trained on summed vectors showed higher performance than systems with averaged vectors, so in the experiments reported below we use the sum of the vectors.

[1]For the complete list of features: `http://www.quest.dcs.shef.ac.uk/quest_files/features_blackbox`

[2]`https://code.google.com/p/word2vec/`

4. Training algorithms

Most word-level QE approaches rely on sequence labelling algorithms. One of the best-performing sequence labelling techniques is **conditional random fields** (CRF) [16], which has been used by many word-level QE systems [17, 18]. However, a CRF model might be less helpful for phrase-level QE. The errors in words may be dependent on each other, and thus the labels of neighbouring tokens can influence each other. Linear chain CRFs are well suited for modelling this type of dependency. However, in phrase-level QE the relatedness of word-level errors is already captured by the phrases. In other words, if the segmentation is accurate, it encapsulates related errors in one unit. While there are no constraints on labels of adjacent phrases (i.e., two or more OK/BAD phrases can occur consecutively), these labels are not expected to be as closely related as those in word-level QE. Therefore, we also explore a standard classifier, a **random forest** classifier [19], which showed good performance in our previous experiments on word-level QE.

5. Experiments

We performed a set of experiments to test how phrase-level systems compare to previous work on word-level QE and to find the optimal parameters for the phrase-level training. We tested performance varying the following parameters:

- Segmentation: target segmentation, source segmentation, target+data segmentation,
- Phrases labelling: optimistic, pessimistic or super-pessimistic,
- Feature set: sentence-level features from QuEst, combined `word2vec` word vectors, both sets of features,
- Models: CRF or Random forest.

We conducted our experiments on two datasets used for the QE shared tasks in 2014 and 2015, so we can compare the performance of our systems with state-of-the-art results.

5.1. Systems

The training of a phrase-level QE system was performed with the open-source QE tool `Marmot`[3]. We trained three distinct systems on three datasets:

- **phrase-wmt-14**: trained on the WMT-14 dataset labelled with error types [10].
- Two systems were trained on fractions of the WMT-15 dataset [13]. This dataset has 11,000 post-edited automatic translations. However, the majority of them contain too few errors, and QE systems trained on the full dataset tend to perform overly optimistic labellings. Therefore, following [20] we use only sentences with the highest HTER score (i.e. largest number of errors normalised by the sentence length):

[3]`https://github.com/qe-team/marmot/tree/phrase_level`

– **phrase-wmt-15-2000**: trained on the 2,000 worst sentences from WMT-15,
– **phrase-wmt-15-5000**: trained on the 5,000 worst sentences from WMT-15.

We also compare our system with the following representative systems that participated in the WMT-14 and WMT-15 QE shared tasks at the word level:

- Systems from WMT-14:
  - **Baseline-all-bad** — trivial baseline strategy that assigns the tag "BAD" to all words. No other system could beat it in terms of F_1-BAD score.
  - **FBK-UPV-UEDIN** [18] — system with features from word posterior probabilities and confusion network descriptors computed over 100,000-best translations. Tagging was done with bidirectional long short-term memory recurrent neural networks. This was the best system in WMT-14.
  - **LIG** [17] — system with 25 black-box features and was trained with CRF. It was the 3rd best system in WMT-14.
- Systems from WMT-15:
  - **Baseline** [13] — system that was used as a baseline at the WMT-15 word-level QE task.
  - **Baseline-all-bad** — the same "all-bad" strategy.
  - **UAlacant** [21] — system that used features drawn from pseudo-references (automatic translations of the source sentence) generated by different MT systems, and baseline features released for the task. Best best-performing system.
  - **Shef-word2vec** [22] — system that used word vector representations as features and performed labelling with a CRF model. This system was ranked 3rd out of 8.

5.2. Tools and datasets

Besides the training and test sets, a QE system requires various resources and tools for feature extraction:

- The word alignment model was trained on the Europarl corpus [23] using the `fast-align` tool[4].
- LM and n-gram count features were extracted using LMs trained on the Europarl corpus using SRILM[5].
- POS features were extracted with TreeTagger [24].
- The translation probability features were computed using lexical probability tables trained with Moses system [11] on the Europarl corpus.
- The word vector representations were computed with `gensim` [25] — Python implementation of `word2vec` models. The training data for the vectors is the concatenation of Europarl, News-commentary[6] and News crawl[7] corpora. The vectors are 500-dimensional.

[4] https://github.com/clab/fast_align
[5] http://www.speech.sri.com/projects/srilm/
[6] http://statmt.org/wmt15/

5.3. Segmentation properties

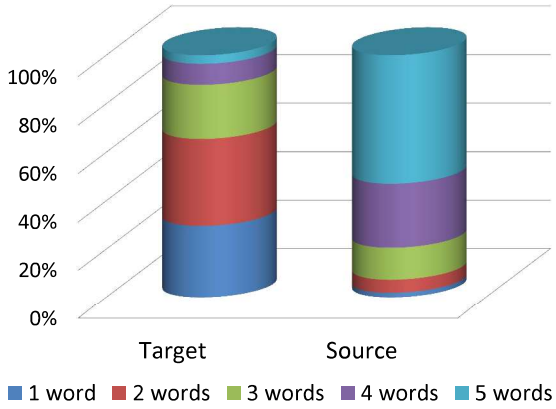

Figure 2: Phrase length frequencies for different segmentation techniques.

The segments produced by two segmentation strategies differ substantially. The main difference is the distribution of phrase lengths: while the **target** segmentation tended to segment the sentences into shorter phrases, the majority of phrases used by the **source** segmentation are 5-word long (see Figure 2). This is explained by the fact that the former strategy uses an independent translation table, whereas the latter decodes the sentences with a translation table trained on the same sentences, so it contains longer phrases.

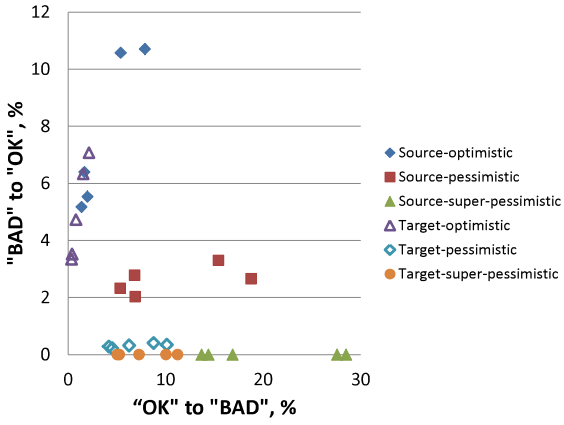

Figure 3: Percentages of word labels modified for datasets segmented with different segmentation techniques (source/target) and re-labelled with either of the labelling strategies (optimistic/pessimistic/super-pessimistic).

We also looked at the amount of word labels that were modified by different labelling strategies under the target-

[7] http://statmt.org/wmt14/

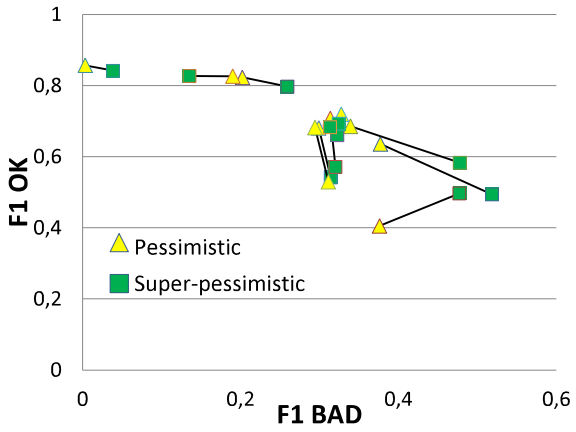

Figure 4: Results for systems with pessimistic and super-pessimistic phrase tagging schemes. Results of systems that differ only in terms of tagging strategy are joined with a line.

based and source-based segmentation types. Figure 3 shows the percentage of words in different datasets that needed to change the label from "OK" to "BAD" and vice-versa. Under source segmentation all labelling techniques become more aggressive, i.e. they change more words. The "optimistic" strategy changes zero or few words from "OK" to "BAD", whereas the "super-pessimistic" strategy does not change words from "BAD" to "OK". Datasets converted with the "pessimistic" strategy contain both types of conversions, but tend to add "BAD" labels rather than "OK" labels.

5.4. Selection of optimal parameters

Here we study which parameters we should use to achieve the best prediction quality for our datasets. We found that most of the parameters depend on datasets and values of other parameters. In addition, the performance of a system is difficult to define: as the F_1 score for the "BAD" class (primary metric for the word-level QE task used for systems comparison in [13]) grows, the F_1 score for the "OK" class drops. In order to account for both of them we plot the F_1-BAD with respect to F_1-OK scores. In each plot we compare systems that differ in one parameter. They are usually shown as items of different colours and shapes. Some items of the same configuration can lie quite far apart. That happens because other parameters of a given pair of systems influenced their performance.

The performance of systems that use different **labelling** schemes follow a certain pattern: the F_1-BAD grows as more negative data is added, while the F_1-OK score drops. Thus, the 'optimistic' labelling scheme is almost always inferior to the other two strategies. The 'pessimistic' and 'super-pessimistic' schemes perform closer, but the latter returns higher F_1-BAD scores for most settings (Figure 4).

This can also be attributed to the source **segmentation** strategy, which generates longer phrases and therefore requires more words to change tag from "OK" to "BAD". Figure 5 shows the comparison of different segmentation strategies and **training algorithms**. It can be seen that CRF produced the best- as well as the worst-performing systems depending on the type of segmentation: the source-segmented data achieves high F_1-BAD score, whereas target segmentation does not perform well in terms of F_1-BAD. On the other hand, the systems trained with the Random Forest classifier do not discriminate between the segmentation types. In addition to that, these systems proved very unstable, whereas CRF always returned the same results for a given configuration. In order to get more meaningful results, we ran the Random Forest classifier 20 times for each configuration and averaged the results.

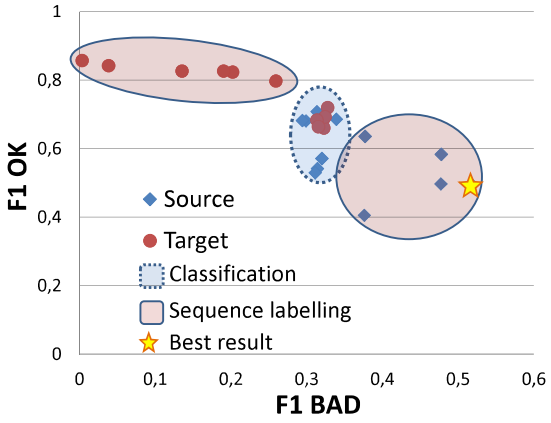

Figure 5: Differences between target and source segmentation and between classification and sequence labelling for phrase-level systems.

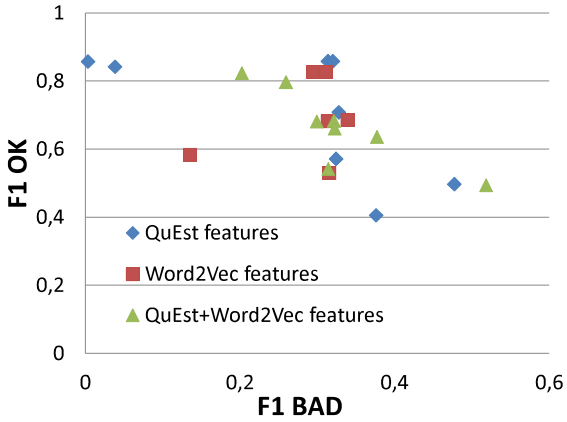

Figure 6: Performance of systems with different feature sets.

The different sets of **features** do not lead to as much variance in performance as the other parameters. However, we can notice that systems with `word2vec` features are more stable and less dependent on other parameters: all systems which use these features perform closely. The use of QuEst and `word2vec` features in combination can lead to the im-

proved performance, whereas systems using only QuEst features are the least stable.

The settings that returned the highest F_1-BAD scores for all the datasets were similar: **source segmentation**, **super-pessimistic labelling**, system trained with **CRF** (see yellow star in Figure 5). The optimal feature sets differ for different datasets. All the figures in this section show the performance of systems trained on 5,000 sentences from the WMT-15 dataset, but the the trends hold for the rest of the systems.

5.5. Comparison to word-level systems

We trained our phrase-level systems on datasets used in the WMT-14 and WMT-15 QE shared tasks, so that we can compare our systems with word-level systems for the task. The WMT-14 system used QuEst features, the WMT-15 system with 2,000 sentences — `word2vec` features, the WMT-15 system with 5,000 sentence — the combination of QuEst and `word2vec` features (although for both WMT-15 systems all feature sets performed closely). The rest of the parameters were fixed for all the datasets: source segmentation, super-pessimistic labelling, CRF.

System	F_1-**BAD** ↑	F_1-OK	Weighted F1
phrase-wmt-14	62.76	39.07	56.80
Baseline-all-bad	52.52	0.0	18.7
FBK-UPV-UEDIN	48.72	69.33	61.99
LIG	44.47	74.09	63.54

Table 1: Performance on WMT-14 test set, systems sorted from best to worst, our system in bold.

System	F_1-**BAD** ↑	F_1-OK	Weighted F1
phrase-wmt-15-5000	51.84	49.38	51.08
phrase-wmt-15-2000	51.57	49.05	50.79
UAlacant	43.12	78.07	71.47
SHEF-word2vec	38.43	71.63	65.37
Baseline-all-bad	31.75	0.0	5.99
Baseline	16.78	88.93	75.31

Table 2: Performance on WMT-15 test set, systems sorted from best to worst, our systems in bold.

Table 1 shows the performance of systems trained and tested on the QE dataset released for the WMT-14 shared task. Our system is the only system which beats the trivial all-bad baseline strategy in terms of F_1-BAD score. The same trend is seen in Table 2, which shows the performance of systems on the WMT-15 data. Both our systems outperform all other system including the winner. They achieve very close scores, which confirms that sentences with less errors do not contribute much for word-level QE.

6. Conclusions and future work

We introduced an approach for quality estimation of MT at the phrase level. To the best of our knowledge, this is the first attempt to label MT phrases with quality. We found that our phrase-level systems outperform word-level systems.

We tested a number of different parameters and found that sentence-level features give better results than word embedding features, CRF model performs better than Random Forest classifier, and the best segmentation strategy is to perform decoding of a source sentence restricting the decoder to output the target sentence, and use the phrase segmentation generated during the decoding. The best tagging strategy is to assume that every phrase that contains at least one "BAD" word should be tagged as "BAD".

In future work we will investigate the performance of other training algorithms. We believe that phrase-level QE can benefit from more advanced algorithms that take into account the segmentation of a sentence in subsequences. For example, Semi-Markov CRFs [26] are designed to solve segmentation and labelling tasks jointly, and higher order CRFs [27] explicitly consider relations between non-adjacent words which can be useful for modelling phrase errors.

An issue with phrase-level QE is that all available datasets are annotated only at the word level. Another direction for future work will thus be the development of a dataset of automatic translations annotated for quality at the level of phrases. From an application perspective, we assume that the phrase segmentation should be guided by segments in statistical MT rather than linguistic properties of the data. However, it would also be interesting to test the usefulness linguistically-informed segmentation.

Finally, further research is necessary to design features that are specific for phrase-level QE. Phrases combine properties of sentences and words: they are sequences, like sentences, but can be quite short, so sentence-level features may be uninformative. The usefulness of linguistically motivated features in particular needs to be tested: as the phrase segmentation performed by an MT decoder does not take into account linguistic information, features indicating whether a phrase is valid based on linguistic information may not suit the task. On the other hand, linguistic information can be useful as it is often unknown to the MT system.

Phrase-level QE of MT is a new field of research. In this paper we proposed the first strategy for the task, highlighted some of its challenges and outlined possible directions of future work.

7. Acknowledgements

This work was supported by the People Programme (Marie Curie Actions) of the European Union's Framework Programme (FP7/2007-2013) under REA grant agreement n° 317471.

8. References

[1] S. Gandrabur and G. Foster, "Confidence estimation for translation prediction," in *HLT-NAACL-2003*, Edmonton, Canada, 2003, pp. 95–102.

[2] N. Ueffing and H. Ney, "Word-Level Confidence Estimation for Machine Translation using Phrase-Based Translation Models," in *HLT-EMNLP-2005*, no. October, Vancouver, Canada, 2005, pp. 763–770.

[3] R. Zens and H. Ney, "N-Gram Posterior Probabilities for Statistical Machine Translation," in *WMT-2006*, no. June, 2006, pp. 72—-77.

[4] L. Specia, N. Cancedda, M. Dymetman, M. Turchi, and N. Cristianini, "Estimating the sentence-level quality of machine translation systems," in *EAMT-2009*, Barcelona, Spain, 2009.

[5] K. Shah, T. Cohn, and L. Specia, "An investigation on the effectiveness of features for translation quality estimation," in *MT Summit XIV*, Nice, France, 2013, pp. 167–174.

[6] O. Bojar, C. Buck, C. Callison-Burch, C. Federmann, B. Haddow, P. Koehn, C. Monz, M. Post, R. Soricut, and L. Specia, "Findings of the 2013 Workshop on Statistical Machine Translation," in *WMT-2013*, Sofia, Bulgaria, August 2013, pp. 1–44.

[7] F. Blain, J. Senellart, H. Schwenk, M. Plitt, and J. Roturier, "Qualitative Analysis of Post-Editing for High Quality Machine Translation," in *MT Summit XIII*, Xiamen, China, 2011, pp. 164–171.

[8] A. Lommel, A. Burchardt, M. Popović, K. Harris, E. Avramidis, and H. Uszkoreit, "Using a New Analytic Measure for the Annotation and Analysis of MT Errors on Real Data," in *EAMT-2014*, 2014, pp. 165–172.

[9] N. Bach, F. Huang, and Y. Al-Onaizan, "Goodness: A Method for Measuring Machine Translation Confidence," in *ACL-2011*, Portland, Oregon, 2011, pp. 211–219.

[10] C. Buck, P. Pecina, J. Leveling, M. Post, L. Specia, and H. Saint-amand, "Findings of the 2014 Workshop on Statistical Machine Translation," in *WMT-2014*, Baltimore, USA, 2014, pp. 12–58.

[11] P. Koehn, H. Hoang, A. Birch, C. Callison-burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation," in *ACL-2007*, Prague, Czech Republic, 2007, pp. 177–180.

[12] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and R. Weischedel, in *AMTA-2006*.

[13] O. Bojar, R. Chatterjee, C. Federmann, B. Haddow, C. Hokamp, M. Huck, V. Logacheva, P. Koehn, C. Monz, M. Negri, P. Pecina, M. Post, C. Scarton, L. Specia, and M. Turchi, "Findings of the 2015 Workshop on Statistical Machine Translation," in *WMT-2015*, Lisbon, Portugal, 2015.

[14] L. Specia, K. Shah, J. G. C. de Souza, and T. Cohn, "QuEst - A translation quality estimation framework," in *ACL-2013*, Sofia, Bulgaria, 2013.

[15] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[16] J. Lafferty, A. Mcallum, and F. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," in *ICML-2001*, 2001, pp. 282–289.

[17] N. Q. Luong, L. Besacier, and B. Lecouteux, "Lig system for word level qe task at wmt14," in *WMT-2014*, Baltimore, USA, June 2014, pp. 335–341.

[18] J. G. Camargo de Souza, J. González-Rubio, C. Buck, M. Turchi, and M. Negri, "Fbk-upv-uedin participation in the wmt14 quality estimation shared-task," in *WMT-2014*, Baltimore, Maryland, USA, June 2014, pp. 322–328.

[19] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[20] V. Logacheva, C. Hokamp, and L. Specia, "Data enhancement and selection strategies for the word-level quality estimation," in *WMT-2015*, Lisboa, Portugal, September 2015, pp. 311–316.

[21] M. Esplà-Gomis, F. Sánchez-Martínez, and M. Forcada, "Ualacant word-level machine translation quality estimation system at wmt 2015," in *WMT-2015*, Lisbon, Portugal, September 2015, pp. 309–315.

[22] K. Shah, V. Logacheva, G. Paetzold, F. Blain, D. Beck, F. Bougares, and L. Specia, "Shef-nn: Translation quality estimation with neural networks," in *WMT-2010*, Lisbon, Portugal, September 2015, pp. 342–347.

[23] P. Koehn, "Europarl: A Parallel Corpus for Statistical Machine Translation," in *MT Summit X*, 2005.

[24] H. Schmid, "Probabilistic part-of-speech tagging using decision trees," in *Proceedings of the International Conference on New Methods in Language Processing*, Manchester, UK, 1994.

[25] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *LREC-2010*, Valletta, Malta, May 2010, pp. 45–50.

[26] S. Sarawagi and W. W. Cohen, "Semi-markov conditional random fields for information extraction," *Advances in Neural Information Processing Systems 17*, pp. 1185—-1192, 2004.

[27] N. Ye, W. S. Lee, H. L. Chieu, and D. Wu, "Conditional Random Fields with High-Order Features for Sequence Labeling," *Journal of Web Semantics*, vol. 2, p. 2, 2004.